


3	2	1	0
7	6	5	4
B	A	9	8
F	E	D	C
+	PC	DA	AD
	RS	ST	GO

# DE KENNERS 14

1	4	0	0	0	0
---	---	---	---	---	---

KIM GEBRUIKERS CLUB NEDERLAND  
4de Jaargang Nr. 14  
20 december 1980


--

--

--

Samenstelling van het bestuur:

Voorzitter : Co Filmer  
Dorpsstr 1051  
1566 JE ASSENDELFT  
Tel.: 075 - 210023

Sekretaris : Anton Müller  
Gijn Semeynsstr 78 1  
1061 GM AMSTERDAM  
Tel.: 020 - 860245

Penningmeester : Ted Schouten  
Junoplns 57  
2024 RM HAARLEM  
Tel.: 023 - 257171  
Postgitorrek.nr.: 3757649

Regeling accommodatie voor KIM-club bijeenkomsten : Bob van de Oudewetering  
Industriewg 12  
2102 LM HEERSTEDDE  
Tel.: 023 - 286444

Technisch adviseur, cassette programma bibliotheek en propaganda KIM-club : Uwe Schröder  
Echternachln 161  
5625 KC EINDHOVEN  
Tel.: 040 - 421821

Software adviseur en regeling programma van KIM-club bijeenkomsten : Sebo Woldringh  
Vlieverink 619  
1104 KC AMSTERDAM ZUIDOOST  
Tel.: 020 - 900085

Organisatie, hardware en beheer KIM-club-KIM : Rinus Vleesch Dubois  
F Nighthinralestr 212  
2037 NG HAARLEM  
Tel.: 023 - 330993

Pagina:

Inhoudsopgave	1
Inhoudsopgave	2
Van het bestuur	3
Van de Redactie	4
JUNIOR H J C Otten	8
One armed bandit for the junior	12
Ervaringen met de Junior (J van Sprang,	16
W L van Pelt)	18
JUNIOR Leestrouine voor Kansas City	26
cassette interface (Sebo Woldringh)	27
JUNIOR Schrijffrouine voor Kansas City	36
cassette interface (Sebo Woldringh)	39
PET Initialisator (R Uphof)	40
System software: RAE printer bootstrap	
KIM Schaakprogramma (T Kortekaas)	
Apple tapes met de KIM (F Mesander en	
H J C Otten)	
Patches op Microsoft BASIC (HJC Otten)	
FLOPPY DISC AANWIJZINGEN	

De KIM KENNER is een uitgave van de KIM gebruikers club Nederland.

Adres voor het in-zenden van en re-akties op artikelen voor de KIM KENNER:

p/a H.J.C. Otten  
Dr Schaepmanstr 15  
1381 BG WEESP  
Tel.: 02940-13349

Redactie KIM KENNER:  
Anton Müller  
Hans Otten  
Peter Visser

Geheel of gedeelte-lijke overname van de inhoud van de KIM KENNER zonder toestemming van het bestuur is verboden.

Toepassen van gepu-bliceerde programma's, hardware etc. is al-leen voor persoonlijk gebruik toegestaan.

© 1980 by KIM Gebruik-kers club Nederland.

Verschijnt vijfmaal per jaar.



KIM - GEBRUIKERSCLUB

#### Forum

Op bepaalde bijeenkomsten wordt in het programma ruimte gereserveerd voor een forum.

#### Doel van een forum

Parate beschikbare kennis en informatie mee te geven in antwoord op de door de leden gestelde vragen. De leden kunnen tijdens het FORUM-gebeuren hun problemen, t.w. hard- en/of software de vrije loop laten. Tevens kunnen ideeën en voorstellen gelanceerd worden. Voorddeel van deze directe communicatie is praktische oplossingen voor de vragensteller, zodat deze sneller vooruit komt. Wederom is gebieden dat er binnen de KIM-club veel kennis aanwezig is (ook voor andere 6502 systemen).

Een levendige discussie kwam op gang bij KIM compatibele JUNIOR, waarbij forumleden Hans en Anton onze leden volstoppen met informatie om KIMprogramma's op de junior te kunnen draaien. Samenstelling forum september was:

Hans Otten - Anton Muller - Rinus Vleesch Dubois - Wim van Gelderen - Uwe Schröder - Sebo Woldringh - Co Filmer.

#### KIM-MARKT VOOR U BEKEKEN

Zoals gewoonlijk wordt er op elke bijeenkomst enige tijd voor de "MARKT" ingeruimd. T.a.v. de normale gang van zaken en spelregels:

1. Aanmelden dient voor de aanvang van de "MARKT" te geschieden, zodat een ieder op volgorde van aanmelding aan de beurt komt.
2. Afhankelijk van het aantal personen die iets hebben aan te bieden, wordt de beschikbare tijd verdeeld in "x" minuten per persoon.
3. Alles wat met micro's, terminals, onderdelen, boeken, tijdschriften, software, of wat hiermede in verband gebracht kan worden (in de ruimste zin van het woord), kan aangeboden worden.
4. De KIM-club cq. Bestuur kan niet verantwoordelijk of aansprakelijk gesteld worden voor o.a.:

- Goede werking van het aangeboden goed.
- Defecten en/of garantie.
- Plaats van herkomst en eventuele gevolgen.
- Afspraken betreffende leveringen, levertijden, betalingen, enz.

Globale opsomming van het gebodene tijdens "meeting september te Heemstede"

Hr. v.d. Burg  
Teletype linten  
Prof. 6502 systemen/printen voor videoterminals/8K  
ramboards op kiformaat/losse chips  
Nummerdisplays 0-9 incl latch-kims + dokumentatie  
Boeken, OSI board, Applecomputers, OKI printers, Macro-assembler, dynamic ramboards  
Printers (gebruikt)  
Hr. Verkooy

H.J.C. Otten

Dit is het vijfde en laatste nummer van de KIM KENNER in 1980. Dit nummer ontvangt U in de maand vol feestdagen december, namens redactie en bestuur wens ik U prettigfeestdagen en de beste wensen voor 1980. Als we zoals gebruikelijk in deze tijd terugkijken op het afgelopen jaar, dan is dat voor de KIM 6502 club een goed jaar geweest. In 1980 is de KIM club uitgegroeid tot een echte 6502 gebruikersclub met een flinke groei in ledental. We hebben vele nieuwe enthousiaste leden met allerlei 6502 systemen. Ook in 1981 zal de KIM club vele activiteiten kunnen ontwikkelen, waarbij ook uw bijdrage welkom en nodig is. Een club als de KIM club wordt geheel gedragen door de leden, waarbij de bijdragen van gevorderden als beginners nodig zijn. Als we allemaal ons best doen dan kunnen we onze interesse, de computer, door gezamenlijke inspanning voor ieder inhoud geven.

Een van de belangrijkste activiteiten van de KIM club is de tweemaandelijksse bijeenkomst. Een belangrijk onderwerp van de bijeenkomst op 17 januari zal de taal FORTH en de implementatie daarvan op allerlei 6502 systemen zijn. We zijn er erg blij mee dat we een bekend deskundige, Dr. Hans Nieuwenhuizen, secretaris van de Europese FORTH gebruikers Groep, bereid hebben gevonden daar een ongetwijfeld leerzame en interessante inleiding in de vorm van een lezing over te willen houden. Na deze voor ieder te volgen lezing zal voor de meer geïnteresseerden een workshop onder leiding van Dr. H. Nieuwenhuizen worden gehouden. In ieder geval kan dan met FORTH op Apple's met een diskdrive worden gewerkt, neem uw Apple gerust mee daarvoor. Later kan FORTH op allerlei 6502 systemen draaien. FORTH is een hogere programmeertaal met uitstekende eigenschappen voor real time controle toepassingen, en stelt daarbij beschikden eisen aan intern en extern geheugen. Ik raad U aan van te voren Byte van augustus 1980 en met name het artikel 'What is FORTH?' te lezen en ook de Byte's van september en oktober over FORTH. Het voordeel van een onafhankelijke club zoals de KIM club is dat we samen sterk staan. Laat daarom uw positieve of negatieve ervaringen met aankoop van hard- en software weten aan uw medeclubleden.

De gelijkenis tussen de KIM en Junior hard- en software is dusdanig groot, dat het zonder gecompliceerde programma-wijzigingen mogelijk is software van de ene aan de andere computer aan te passen. Daarbij is echter documentatie nodig van die andere computer.

KIM bezitters (en ook andere 6502 gebruikers) zou ik aanraden de Junior boeken te kopen, vooral deel 2 met de Junior monitor is goed leesbare en interessante literatuur over de 6502 en natuur-lijk de Junior.

Mijn eerste indruk van de Junior was dat het een kopie van de KIM was. Wat betreft de hardware is dit ook zonder meer waar, er is één detail waarin de Junior en de KIM duidelijk verschillen en dat is de wijze van aansturen van de LED display's. Het lijkt precies op elkaar, alleen de polariteit is omgedraaid. Als we bij de KIM een één naar een LBD-segment sturen, dan moet bij de Junior een nul worden gezonden. Verder is de werking identiek.

Dit verschil uit zich duidelijk in de hex-naar zevenssegment tabel, de tabel in de Junior (LF0F en verder) is gelijk aan die van de KIM (LF07 en verder) maar dit voor dit geïnverteerd!

Junior gebruikers kunnen een grote hoeveelheid programma's vinden in het "First book of KIM", maar stuiten dan op onbekende KIM routines, waarvoor ze de KIM manuals moeten raadplegen.

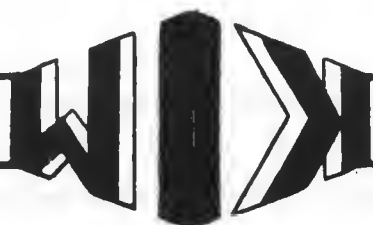
Om de conversie te ondersteunen zal ik een aantal overeenkomstige KIM en Junior routines aanwijzen die veel gebruikt zullen worden. Hieruit blijkt dat de software ook grote overeenkomsten vertoont, een aantal subroutines heeft dezelfde naam en coding. Er zijn echter een aantal opmerkelijke verschillen in de software.

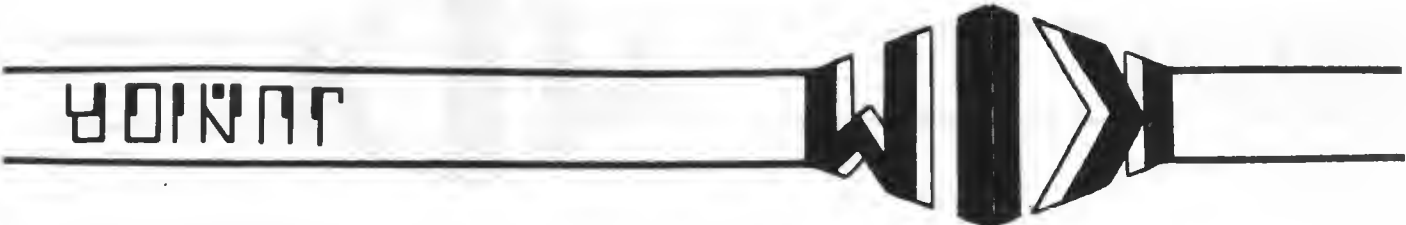
De KIM heeft meer I/O mogelijkheden zoals een teletype en een audio cassette recorder, voorzieningen die bij de Junior ongetwijfeld zullen komen. Een raadgeving: wacht op de ontwikkelingen bij Elektron en ga niet zelf ontwikkelen of kopen, daar krijgt u spijt van. Doe zoveel mogelijk gemeenschappelijk!

H.J.C. Otten

Conversie van KIM naar JUNIOR en omgekeerd

JUNIOR





De Junior heeft een erg leuk uitgevoerde monitor wat betreft het invoeren en uitproberen van programma's met het toetsenbord en de display's, wat in de KIM niet aanwezig is .

Overeenkomstige routines

KIM JUNIOR

1. \$1C00 - SAVE \$1C00 - SAVE

Bewaar registers op  
zero page en start monitor  
Zelfde , met zelfde zero page  
lokaties etc

2. \$1C1D - RESET \$1C22 - RESET

zet poorten goed en  
start monitor , na  
gelijk aan KIM

3. \$1F19 - SCAND \$1D88 - SCAND

toon dat  
op display van FA,FB  
loopt over in AK  
gelijk KIM

4. \$1F1F - SCANDS \$1D8E - SCANDS

toon inhoud van  
FA,FB,9 op display als  
3 x 2 hex getallen  
loopt over in AK  
worden getoond ( 1,2 of 3 ) , KIM  
\$00F6 ( BYTES ) bepaalt hoeveel bytes  
gelijk aan KIM , maar inhoud

5. \$1C77 - START \$1C33 - START

start monitor , wachten  
op toetsdruk  
analoog

6. \$1F6A - GETKEY \$1DF9 - GETKEY

haal toets van keyboard  
A = toetsnummer , A > 15  
ongjuiste toets indruk  
zelfde functie, toetsen zelfde  
nummering als bij KIM

7. \$1FEF - AK \$1DB1 - AK

Bepaal of toets ingedrukt  
A ≠ 0 dan ingedrukt  
gelijk

De KIM heeft RAM vanaf adres 0000 t/m 03FF evenals de Junior.  
Daarnaast heeft de KIM RAM ( van de 6530's ) van 1780 t/m 17FF.  
De Junior heeft RAM van 1A00 t/m 1A7F, ook 128 bytes

#### RAM Lokaties

Interrupt vector is bij de KIM de RAM lokatie 17FE en 17FF,  
bij de Junior 1A7E en 1A7F.  
Non maskable vector is bij de KIM 17FA en 17FB, bij de Junior  
is dit 1A7A en 1A7B.  
De break vector ( instructie BRK ) is bij de KIM de interrupt  
vector 17FE etc, bij de Junior 1A7E etc.

#### Vectoren

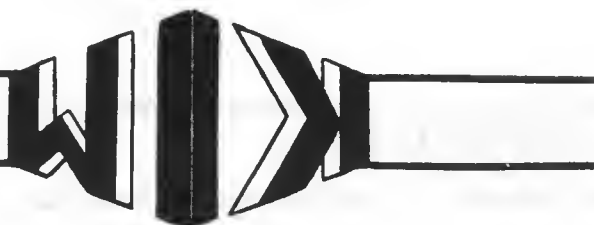
De 6530 en 6532 hebben gelijke timers, hiervoor is het beste de  
voor iedere serieuze 6502 gebruiker onontbeerlijke Program en  
HardwareManuals van de fabrikant te raadplegen.  
Timerlokaties 1704 etc van de KIM ontbreken bij de Junior.  
Overeenkomstige timers zijn de KIM lokaties 1744 t/m 174F en  
de Junior lokaties 1AF4 t/m 1AFF ( bv CNPA Junior 1AF4 = KIM  
C1K1F 1744 etc ) .  
De edge detect mogelijkheid van de Junior ontbreekt bij de KIM.

#### Timers

Bij de KIM hebben we twee PIA's in de twee 6530 IC's. In de  
Junior zit er een in de 6532. Ze gedragen zich gelijk, die 6530  
en 6532 als PIA, maar ze zitten op andere lokaties.  
KIM PIA 1700-1703 ontbreekt bij de Junior.  
KIM PIA 1740-1743 = Junior PIA 1A80-1A83, vrijwel gelijk gebruikt.  
PAD KIM is bijvoorbeeld PAD Junior : 1740 = 1A80.

#### PIA Lokaties

JUNIOR





GETBYT routine van de JUNIOR

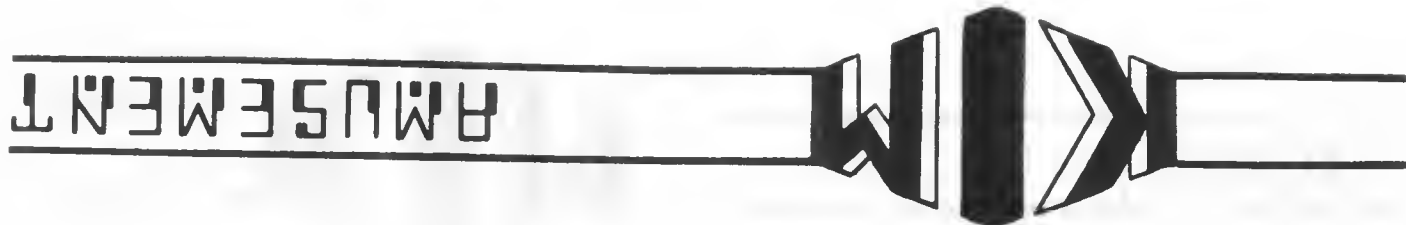
In de Junior zit een routine, GETBYT genaamd, die twee toets-indrukken afwacht en ze combineert tot een hex getal in de accumulator. Zijn er geen hex toetsen ingedrukt ( 0 .. F ) dan is de N flag 0, anders 1. In de Junior boeken wordt dankbaar gebruik gemaakt van deze routine, onder andere in de branch berekenings routineBRANCH = LFD5 ( overigens is deze routine een bewerkte versie uit het First book of KIM ; ). Voor geïnteresseerden herhaal ik deze JUNIOR GETBYT routine uit het Junior boek 2 van Elektuur : ( in assembler notatie ; )

```

GETBYT JSR SCANA    read high order nibble
        CMPIM $10
        BPL BYTEND  command key ?
        ASIA
        ASIA
        ASIA    if not save high order nibble
        ASIA
        STA NIBBLE ( bv 00FE )
        JSR SCANA
        CMPIM $10
        BPL BYTEND  command key ?
        ORA NIBBLE if not , compose byte
        LDIM      set N=1
        RTS
BYTEND  JSR SCANDS display
        BNE SCANA   key released ?
        SCANB      JSR SCANDS display
        BEQ SCANB   any key
        JSR SCANDS display
        BEQ SCANB   any key still depressed ?
        JSR GETKEY if yes , return with key in accu
        RTS

```

Tot zover de conversie hulp . Heeft U andere conversie problemen stuur me dan een briefje . Heeft U met succes iets omgezet , laat het dan aan de KIM KENNER weten voor Uw medegebruikers ..



\*\*\*\*\*

ONE ARMED BANDIT FOR THE JUNIOR MICRO COMPUTER

ORIGINALLY A KIM-I VERSION, MODIFIED BY

ANTON MUELLER

SING SEMINSTR 78 1

1001 GN AMSTERDAM

THE NETHERLANDS

IN ORDER TO MAKE IT SUITABLE FOR THE JUNIOR

THE PROGRAM INTERFACES WITH THE JUNIOR KEY-

BOARD AND DISPLAYS TO PRODUCE A TONE

ARMED BANDIT TYPE OF SLOT MACHINE ACTIVITY

THE RIGHT-HAND SIDE OF THE DISPLAYS SHOWS

THE PLAYER'S BALANCE IN DUTCH FLORINS.

AN EARLIER VERSION OF THIS PROGRAM ALLOWS

THE WHEELS TO SPIN UNTIL THEY ARE STOPPED,

ONE AT A TIME, BY THE PLAYER TOUCHING A KEY.

WHILE THE EFFECT WAS GOOD, THEY LED TO

EXTREMELY HEAVY USAGE OF THE PUSHBUTTONS.

THIS VERSION CAUSES THE WHEELS TO STOP AUTO-

MATICALLY.

MUCH OF THE PROGRAM IS DEVOTED TO PRODUCING

THE FEEL OF THE SIMULATED MACHINE. IT

WOULD BE VERY EASY, OF COURSE, TO HAVE THE

WHEELS START AND STOP INSTANTANEOUSLY, AND

THE CASH BALANCE DISPLAYED IMMEDIATELY;

BUT THIS WOULD NOT RESEMBLE THE REAL

MECHANICAL MACHINE.

THERE IS NO HOUSE PERCENTAGE, SO THE

PLAYER CAN GO FOR CONSIDERABLE TIME ON HIS

INITIAL 25 GUILDERS. HOWEVER, HE IS NOT ALLOWED

TO WIN A TOTAL OVER 99 GUILDERS, AND THE

WHEELS WILL NOT SPIN IF HE LOSES ALL HIS

MONEY.

THE PROGRAM STARTS AT LOCATION 0200.

SPIN THE WHEELS BY TOUCHING ANY BUTTON

BETWEEN 0 AND F.

AUTHOR: F. J. BUTTERFIELD

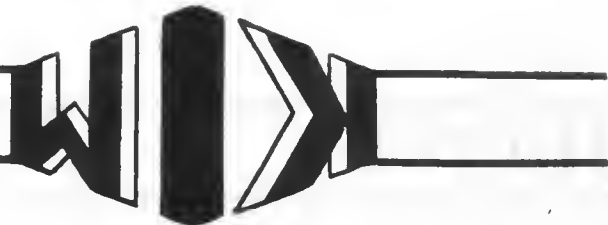
14 BROOKLYN AVENUE

TORONTO - ONTARIO M4M2X5

C A N A D A

Address	Disassembly	Comment
0310	*	WORK AREA'S IN PAGE ZERO
0320	*	
0330	BANDIT ORG \$0200	
0340	WINDOW *	
0350	AMT *	
0360	ARROW *	
0370	RWD *	
0380	STALL *	
0390	TUMBLE *	
0400	+	
0410	+	REFERENCES TO JUNIOR MONITOR
0420	*	
0430	SCDSB *	
0440	FADD \$1A81	
0450	FAD \$1A80	
0460	FBD \$1A82	
0470	LOOK *	
0480	*	
0490	+	
0500	+	MAIN PROGRAM STARTS HERE
0510	GO	
0520	LDAM #25	ASSUME THE PLAYER
0530	STA AMT	STARTS WITH 25 BUCKS
0540	USR CVANT	CHANGE TO DISPLAY
0550	LDAM #00	
0560	STA ARROW	
0570	*	
0580	020B 20 BD 02	LFA
0590	020E D0 FB	EME LFA
0600	0210 E6 07	ROLL
0610	0212 20 BD 02	USR DISPLAY
0620	0213 F0 F7	BEA ROLL
0630	0217 A7 03	LDAM #03
0640	0219 85 06	STA ARROW
0650	021B F8	SED
0660	021C 38	SED
0670	021D A5 05	LDA AMT
0680	021F E9 01	SECM #01
0690	0221 85 05	STA AMT
0700	0223 20 BA 02	USR CVANT
0710	0226 26 09	ROL TUMBLE
0720	0228 20 BD 02	USR DISPLAY
0730	022B C6 08	DEC STALL
0740	022F A6 06	LDX ARROW
0750	0231 A5 09	LDA TUMBLE
0760	0233 09 F7	ORAIN #F7
0770	0235 29 BF	ANDIN #BF
0780	0237 95 01	STAX WINDOW +01 TO DISPLAY
0790	0239 46 09	LSR TUMBLE
1000	023B 46 09	LSR TUMBLE

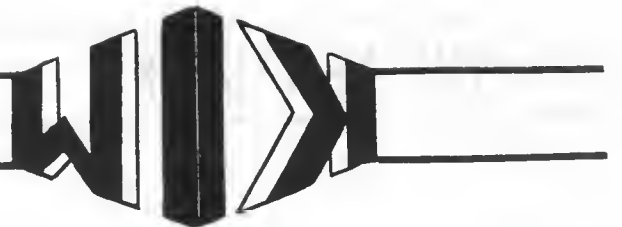
A stylized graphic of a film strip. The word "MUSEMENT" is written vertically in a bold, sans-serif font along the length of the film strip. The film strip has a black border and a white center. The letters are black and oriented vertically. The film strip is shown as a continuous loop, with the end of the strip curving back to the beginning. The overall design is minimalist and modern.



SYMBOL TABLE 3800 38A8

AMT	0005	ARROW	0006	BANDIT	0200	CVANT	02BA
DISPLY	028D	GO	0200	INDIS	0293	LITE	029F
LOK	0284	LOOK	1F0F	LFA	020B	LFB	0228
LPC	0264	NOMAT	027E	OVER	0291	P	02D1
PADD	1A81	PAD	1A80	PAY	0260	PAY	02CE
PBD	1A82	ROLL	0210	RWD	0007	SCDSB	1DA0
STALL	0008	TUMBLE	0009	WINDOW	0000	ZIP	02AA

1510:	0296	A9	7F	LDAIN	#7F	STA	PADD	LDAIN	#0B	LDAIN	#04	LDAAX	WINDOW	STA	PBD	STA	PAD	CLD	LDAIN	#7F	SECIN	#01	BNE	ZIP	STA	PBD	INX	INX	INX	DEX	BPL	LITE	USR	SCDSB	RIS	AMOUNT CONVERSION	CVANT	1710:	02BA	A0	03	1720:	02BA	A0	03	1730:	02BC	Z7	0F	1740:	02BE	AA	TAX	LDAAX	LOOK	STA	WINDOW	LDA	AMT	1750:	02C6	4A	1760:	02C2	83	00	1770:	02C4	A0	03	1780:	02C6	4A	1790:	02C7	4A	1800:	02C8	4A	1810:	02C9	4A	1820:	02CA	AA	1830:	02CB	BD	0F	1F	1840:	02CE	83	01	1850:	02D0	60	RIS
-------	------	----	----	-------	-----	-----	------	-------	-----	-------	-----	-------	--------	-----	-----	-----	-----	-----	-------	-----	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-----	-------	-----	-------------------	-------	-------	------	----	----	-------	------	----	----	-------	------	----	----	-------	------	----	-----	-------	------	-----	--------	-----	-----	-------	------	----	-------	------	----	----	-------	------	----	----	-------	------	----	-------	------	----	-------	------	----	-------	------	----	-------	------	----	-------	------	----	----	----	-------	------	----	----	-------	------	----	-----



J. van Sprang

Tulp 71

Krimpen a.d. IJssel

W.L. van Pelt

Jacob Jordaanstr. 15

Krimpen a.d. IJssel

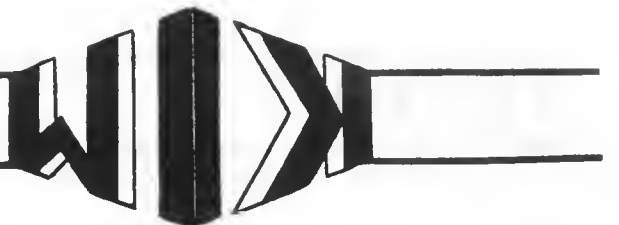
Ervaringen met de JUNIOR

De ervaringen, wensen en verzuchtingen die in een vorige KIM KENNER zijn vermeld vormen niteraad geen afgerond geheel. Vandaar deze

voortzetting.

Bemerkt is dat door ook deze publiekatie contacten zijn ontstaan en versterkt. Inmiddels hebben bovengenoemde elkaar gevonden in een gemeenschapelijk probleem: de behoefte om programma's op de band te zetten, zodat een hoop toetswerk eenmalig kan blijven. Vanwaar die behoefte, als er nog geen JUNIOR-programma's zijn verschenen? Dat bleek niet zo'n nood, want via de sekretaris verkregen we inzicht in het verschil tussen KIM en JUNIOR monitor routines. Niet dat daarmee alles kan worden vertaald in JUNIOR, maar een paar problemen over en weer gegevens worden uitgewisseld. Wij hopen dat het bestuur die verschillen, maar dan wel alle, tussen KIM en JUNIOR, in de KIM KENNER zullen publiceren, zodat een ieder daar profiteert van heeft. (Is reeds gebeurd. Zie artikel in dit nummer van Hans Otten; Red.) Gezien onze behoefte werd, zodra de mogelijkheid daar was en in weerwil van de raad die ons gegeven was, een cassette-interface bij Muscript Computer Products aangeschaft. (Deze firma is ook bekend onder de naam CHIP. Red.). Deze interface leverde wel wat stof op. Nu het uiteindelijk werkt, zeg je dat de interface op zich in orde is. Maar om het ding echt aan de praat te krijgen hebben beginnelingen wel een helpende handje nodig in de vorm van een programma. Daar had MCP voor gezorgd, zoals ze al eerder inzag dat de concurrentie toch van je vergt dat de JUNIOR computer beter verkocht kan worden met bedrukte/gegraveerde toetsen dan met die afgrijpsel-ke plakcijfers en -letters en zoals je beter kunt aanrijzen met gratis programma's. Niet al die programma's zijn uitgetroefd, maar zeker is, dat ze niet allemaal werken. Voor wat het cassette-interface programma betreft: er is vele malen telefonisch contact gezocht met MCP en als je iemand kreeg met meer gezag dan de telefoniste, dan nog blijft dat we nu nog op een beter programma wachten. Wat we hebben ontvangen bij de interface bleek goede voeding voor de prullen-

mand. Uiteindelijk heeft de software-deskundige van de KIM club, Sebo Woldringh, ons binnen een kwartier uit de nood geholpen met nieuwe programmatuur voor de cassette in- en uitvoer. Daar moest

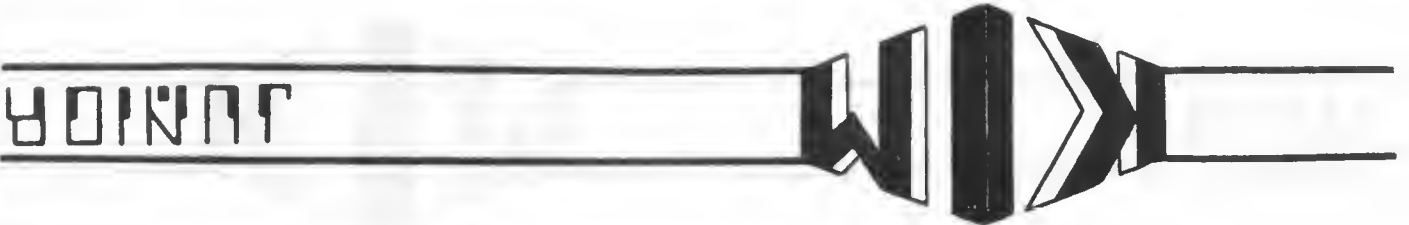


nog wel een kleinigheid aan veranderd worden, maar dat was in twee avonden rond. Dat ging dus duidelijk sneller dan bij MCP. De programma's vindt u hierbij. Nog even wat over de interface. Als u niet in het bezit bent van dit ding: over een paar maanden komt Elektuur met een cassette interface die veel meer belooft. Mocht u de MCP cassette interface toch willen kopen of al bezitten, dan zult u bemerken dat één der IC's is afgeschrapt en in de beschrijving een "Speciaal IC volgens fabrieksspecificaties" heeft. Wat er voor speciaals is aan een doorgewone 74123 flip-flop is ons niet bekend. De interface werkt met TTL-IC's waarvan bekend is, althans zo heb ik begrepen, dat deze een spanning van 2,4 Volt en hoger leest als "1" en 0,8 Volt en lager als "0". Kijkt u naar de technische gegevens, dan blijkt dat de audio-input maar liefst 1,5 Watt nodig heeft. Het atregelen van de interface kon gelukkig bij Van Sprang gebeuren die een praktische ervaring heeft met elektronica. Hij beschikte over een draagbare radio/cassette combinatie. Enige modificaties hieraan en klaar is Kees. Bij mij ging het wat anders. Aangemoedigd door een artikel van Hans Otten in Radio Bulletin nr. 3 van 1980, kocht ik bij Radio Service Twenthe in Den Haag de aanbiddings-inbouw-cassette-deck type Touring 108 ad f 32,50. Het ding wordt in een doos overhandigd en je gaat blij naar huis. Nu is het te hopen dat niet iedereen het overkomt dat bij thuis-komst, zoals bij mij, blijkt dat het meest essentiële onderdeel ontbreekt: de motor. Voor mij werd het door de niet geringe benzinekosten dus geen koopje. Enfin, de vordering die Hans Otten vermeldde is rond gekomen. Hier bleek ook de noodzaak van samenwerking met anderen. Dat lag niet aan het schema, maar aan gebrek aan kennis en ervaring mijnerzijds. Je bent een leek, of niet. Ook de beide versterkers zijn gebouwd. Aangezien uiteindelijk gekozen is voor de universele luidspreker/versterker uit de Elektuur-Halbleitdergids 1980, hangt de eindversterker van Hans Otten met 2,5 Watt nu aan het wachtrek. En zo ook de 1.10 maal versterker. Waarom dat? Wel, gesteld dat de gestandaardiseerde uitgangsspanning van recorders 200 mV is, zoals wij menen, dan bereiken we dus bij 10 maal zo'n 2 Volt. En dat is te weinig voor de TTL-IC's van onze interface, zoals hiervoor vermeld; De Elektuurnut kan zo'n 200 maal versterken, dus daar waren dan ook geen problemen mee.

Nu we stad en land niet meer hoeven af te bellen, kunnen we eindelijk in deel 2 van het JUNIOR-boek duiken om daar te zien dat er inderdaad heel wat in de monitor zit. De behoefte aan uitbreiding zal na het lezen van dit deel wel gloreu.

- - -

N.B.: De 2½ bit tijd op locatie 0148 blijkt soms beter te werken als je die op 24 (hexadecimal 18) zet.



0010:	0100	READ	ORG	\$0100	
0020:					
0030:					*****
0040:					*
0050:					LEES ROUTINE VOOR KANSAS CITY CASSETTE
0060:					INTERFACE VOOR DE JUNIOR MICRO COMPUTER
0070:					FOOTJE 21 VAN DE APPLICATIE CONNECTOR
0080:					DOORVERBINDEN MET DE CASSETTE INPUT
0090:					START ADRES INTIKKEN OP SAL/SAH
0100:					EIND ADRES + 1 INTIKKEN OP EAL/EAH
0110:					*
0120:					AUTEUR: SEBO WOLDRINGH
0130:					----- KLIEVERINK 619
0140:					AMSTERDAM ZUIDOOST
0150:					*
0160:					COPYRIGHT (C) BY S. T. WOLDRINGH
0170:					*
0180:					*****
0190:					CONSTANTES
0200:					-----
0210:	0100	HALF	*	\$0005	(HALVE BIT TIJD = 5)
0220:	0100	HEEL	*	\$000A	(HELE BIT TIJD = 10)
0230:	0100	TWEEH	*	\$0019	(TWEE EN EEN HALVE BIT TIJD = 25)
0240:	0100				
0250:					PAGE ZERO LOCATIES
0260:					-----
0270:	0100	INL	*	\$00F8	
0280:	0100	INH	*		+01
0290:	0100				
0300:					PIA POORT B VAN 6532
0310:					-----
0320:	0100	PBD	*	\$1A82	PIA B DATA REGISTER
0330:	0100	PBDD	*		+01 PIA B DATA DIRECTION REGISTER
0340:	0100				
0350:					GEBRUIKTE TIMER LOCATIE:
0360:					-----
0370:					CNTD
0380:	0100	*		\$1AF7	1024 TIMER
0390:					JUNIOR MONITOR ROUTINES
0400:					-----
0410:	0100	RESET	*		\$1CID
0420:					
0430:					
0440:	0100	AD	59	01	LDA
0450:	0103	85	F8		STA
0460:	0105	AD	5A	01	LDA
0470:	0108	85	F9		STA
0480:	010A	A9	00		LDAIM \$00
0490:	010C	8D	83	1A	STA
0500:	010F	20	53	01	READA
					USR
					READI
					LEES EEN BIT
					OP INPUT
					ZET DATA DIRECTION REGISTER B
					EN ZET DIE IN PAGE ZERO START ADR HIGH
					HAAL START ADRES HIGH OP
					STA
					INL
					EN ZET DIE IN PAGE ZERO START ADR LOW
					HAAL START ADRES LOW OP
					SAL



JUNIOR

```

0510: 0112 D0 FB BNE READA BLIJF Wachten tot er een "0" komt
0520: 0114 A9 05 LDAlM HALF ZET HALVE BIT TIJD
0530: 0116 20 4B 01 READB JSR EN HAAL EEN BIT OP
0540: 0119 D0 F4 BNE READA BLIJF Wachten tot er wat komt
0550: 011B A0 08 LDYIM $08 ZET AANTAL BITS PER BYTE
0560: 011D 48 PHA OP DE STACK
0570: 011E A9 0A READC LDAlM HEEL ZET HELE BIT TIJD
0580: 0120 20 4B 01 JSR EN HAAL EEN BIT OP
0590: 0123 18 CLC DE CARRY FLAG WORDT GEZET
0600: 0124 F0 01 BEQ READD ALS WE EEN "1" HEBBEN
0610: 0126 38 SEC GELEZEN
0620: 0127 68 PLA SCHUIF DE GELEZEN
0630: 0128 2A ROLA BITS IN DE ACCUMULATOR
0640: 0129 48 PHA
0650: 012A 88 DEY VERLAAG LOOP COUNTER
0660: 012B D0 F1 BNE READC
0670: 012D 68 PLA IN ACC STAAT NU EEN CHARACTER
0680: 012E 91 F8 STAlY INL DAT WE STOREN OP
0690: 0130 E6 F8 INC INL HET AANGEGEVEN
0700: 0132 D0 02 BNE READD GEINDEXEERDE
0710: 0134 E6 F9 INC INH ADRES
0720: 0136 A5 F8 READE LDA ALS HET START
0730: 0138 CD 5B 01 CMP EAL ADRES GELIJK IS
0740: 013B D0 0A BNE READD GEMORDEN AAN
0750: 013D A5 F9 LDA INH HET EIND ADRES+1
0760: 013F CD 5C 01 CMP EAH DAN ZIJN WE KLAAR
0770: 0142 D0 03 BNE READD EN GAAN WE TERUG
0780: 0144 4C 1D 1C JMP RESET NAAR DE RESET ROUTINE
0790: 0147 A9 19 READF LDAlM TWEEH ZET 2 1/2 BIT TIJD
0800: 0149 D0 CB BNE READB
0810: 014B 8D F7 1A READG STA 1K TIMER TIJD
0820: 014E 2C F7 1A READH BIT LAAT TIMER
0830: 0151 10 FB BPL READH AFTELLEN
0840: 0153 AD 82 1A READI LDA LEES POORT B (PB6)
0850: 0156 29 40 ANDIM $40 ZET ONGEMENSTE BITS UIT
0860: 0158 60 RTS TERUG NAAR HOOFD ROUTINE
0870: 0159 00 SAL = $00 START ADRES LOW
0880: 015A 00 SAH = $00 START ADRES HIGH
0890: 015B 00 EAL = $00 EIND ADRES LOW (EIND ADRES + 1)
0900: 015C 00 EAH = $00 EIND ADRES HIGH

```

SYMBOL TABLE 3800 388A

```

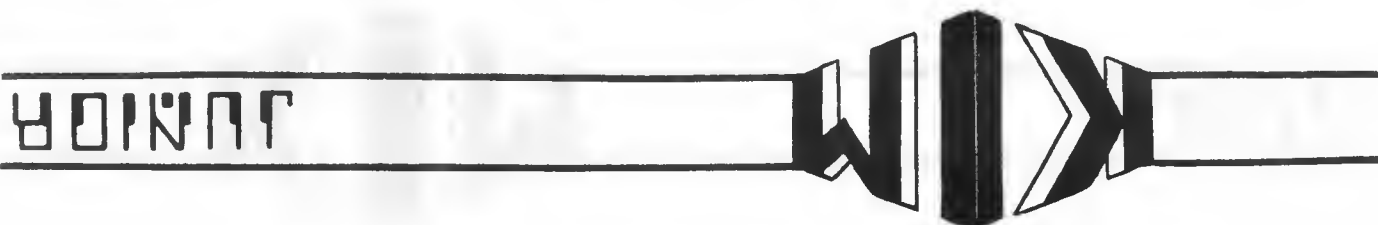
CNTD 1AF7 EAH 015C EAL 015B
HEEL 000A INH 00F9 INL 00F8 PBDD 1A83
PBD 1A82 READ 0100 READA 010F READB 0116
READC 011E READD 0127 READE 0136 READF 0147
READG 014B READH 014E READI 0153 RESET 1C1D
SAH 015A SAL 0159 TWEEH 0019 ICID 0005

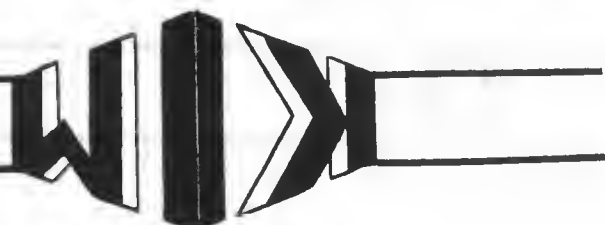
```

```

0010: 1A00 WRITE ORG $1A00
0020:
0030: *****
0040: *
0050: * SCHRIJF ROUTINE VOOR KANSAS CITY CASSETTE
0060: * INTERFACE VOOR DE JUNIOR MICRO COMPUTER
0070: * POOTJE 20 VAN DE APPLICATIE CONNECTOR
0080: * DOORVERBINDEN MET DE CASSETTE OUTPUT
0090: * START ADRES INTIKKEN OP SAL/SAH
0100: * EIND ADRES + 1 INTIKKEN OP EAL/EAH
0110: *
0120: * AUTEUR: SEBO WOLDRING
0130: * ----- KLIEVERINK 619
0140: * AMSTERDAM ZUIDOOST
0150: *
0160: * COPYRIGHT (C) BY S. T. WOLDRING
0170: *
0180: *****
0190:
0200:
0210:
0220: 1A00 HALF * $0005 (HALVE BIT TIJD = 5)
0230: 1A00 HEEL * $000A (HELE BIT TIJD = 10)
0240: 1A00 TWEEH * $0019 (TWEE EN EEN HALVE BIT TIJD = 25)
0250:
0260: PAGE ZERO LOCATIES
0270:
0280: 1A00 INL * $00F8
0290: 1A00 INH *
0300:
0310: PIA POORT B VAN 6532
0320:
0330: 1A00 PBD * $1A82 PIA B DATA REGISTER
0340: 1A00 PBD * PBD +01 PIA B DATA DIRECTION REGISTER
0350:
0360: GEBRUIKTE TIMER LOCATIE:
0370:
0380: 1A00 CNTD * $1AF7 1024 TIMER
0390:
0400:
0410:
0420: 1A00 RESET * $1C1D
0430:
0440: 1A00 AD 56 1A LDA SAL
0450: 1A03 85 F8 STA INL
0460: 1A05 AD 57 1A LDA SAH
0470: 1A08 85 F9 STA INH
0480: 1A0A A9 20 LDAIM $20
0490: 1A0C 8D 83 1A STA PBD
0500: 1A0F A0 00 WRITEA LDYIM $00
ZET INDEX REGISTER Y OP NUL
0440: 1A00 AD 56 1A LDA SAL
0450: 1A03 85 F8 STA INL
0460: 1A05 AD 57 1A LDA SAH
0470: 1A08 85 F9 STA INH
0480: 1A0A A9 20 LDAIM $20
0490: 1A0C 8D 83 1A STA PBD
0500: 1A0F A0 00 WRITEA LDYIM $00
HAAL START ADRES LOW OP
EN ZET DIE IN PAGE ZERO START ADR LOW
HAAL START ADRES HIGH OP
ZET DATA DIRECTION REGISTER
VOOR OUTPUT
ZET INDEX REGISTER Y OP NUL

```





0510:	1A11 B1 F8	LDAI INL	HAAL EEN BYTE OP
0520:	1A13 20 2D 1A	JSR	WRITEC EN SCHRYF HEM WEG
0530:	1A16 E6 F8	INC	VERHOOG
0540:	1A18 D0 02	BNE	WRITEB START ADRES
0550:	1A1A E6 F9	INC	MET 1
0560:	1A1C A5 F8	LDA	ALS HET START
0570:	1A1E CD 58 1A	CMP	ADRES GELIJK IS
0580:	1A21 D0 EC	BNE	WRITEA GEMORDEN AAN
0590:	1A23 A5 F9	LDA	HET EIND ADRES +1
0600:	1A25 CD 59 1A	CMP	DAN ZIJN WE KLAAR
0610:	1A28 D0 E5	BNE	WRITEA EN GAAN WE TERUG
0620:	1A2A 4C 1D 1C	JMP	NAAR DE RESET ROUTINE
0630:	1A2D 18	WRITEC CLC	STOP DE TE SCHRYVEN
0640:	1A2E 48	PHA	BYTE OP DE STACK
0650:	1A2F 20 43 1A	JSR	WRITEC SCHRYF START BIT WEG
0660:	1A32 68	PLA	HAAL TE SCHRYVEN BYTE OP
0670:	1A33 A0 08	LDAIM \$08	ZET AANTAL BITS PER BYTE
0680:	1A35 0A	WRITEC ASLA	SCHUIF EEN BIT IN DE CARRY FLAG
0690:	1A36 48	PHA	ZET HET RESTANT OP DE STACK
0700:	1A37 20 43 1A	JSR	WRITEC SCHRYF EEN BIT WEG
0710:	1A3A 68	PLA	HAAL RESTANT TERUG VAN DE STACK
0720:	1A3B 88	DEY	HERHAAL DIT TOTDAT WE 8 BITS
0730:	1A3C D0 F7	BNE	WRITEC HEBBEN WEGGESCHREVEN
0740:	1A3E 38	SEC	SCHRYF TWEE
0750:	1A3F 20 43 1A	JSR	WRITEC STOP BITS
0760:	1A42 38	SEC	WEG
0770:	1A43 A2 00	WRITEC LDIM \$00	SCHRYF EEN
0780:	1A45 90 01	BCC	WRITEC BIT NAAR
0790:	1A47 CA	DEX	P1A POORT B
0800:	1A48 8E 82 1A	WRITEC STX	FBD BIT 5 (FBS)
0810:	1A4B A9 0A	LDAIM	HEEL ZET HELE BIT TIJD
0820:	1A4D 8D F7 1A	STA	CNTD VOOR TIMER
0830:	1A50 2C F7 1A	WRITEC BIT	CNTD EN LAAT TIMER
0840:	1A53 10 FB	BPL	WRITEC AFLOPEN
0850:	1A55 60	RTS	GA TERUG NAAR HOOFD ROUTINE
0860:	1A56 00	SAL	=
0870:	1A57 00	SAH	=
0880:	1A58 00	EAL	=
0890:	1A59 00	EAH	=

SYMBOL TABLE 3800 387E

CNTD	1AF7	EAH	1A59	EAL	1A58	HALF	0005
HEEL	000A	INH	00F9	INL	00F8	PDDB	1A83
FBD	1A82	RESET	1C1D	SAH	1A57	SAL	1A56
TWEEH	0019	WRITE	1A00	WRITEA	1A0F	WRITEB	1A1C
WRITEC	1A2D	WRITEC	1A35	WRITEE	1A43	WRITEF	1A48
WRITEG	1A50						

De te gebruiken USER PORT bestaat uit een 8-bits databus welke is aangesloten op de data A uitgang van een 6522 VIA (Output Register A) Het genoemde ORA wordt bij een reset van de CBM "hardware" geïnitia- liseerd, dus door het laag worden van de reset van het VIA. Het ORA heeft adres \$E84F en bevat dan \$FF. Het is rechtstreeks geschikt voor input "active low", zolang het eveneens door de reset op 0 ge- initialiseerde DATA DIRECTION REGISTER A (DDRA) in deze toestand blijft. Het DDRA heeft in de CBM \$E843 als adres. Door een bit van het DDRA 1 te maken zal het overeenkomstig bit van het ORA nul worden en geschikt zijn voor output (active high) Eenmaal op 1 gezet zal dat bit dan toch weer als active-low input kunnen dienen, zonder dat men iets aan de inhoud van DDRA hoeft te doen.

## DE CBM USERPORT

In de +6V-leiding van de motor is een weerstand van 2,2 ohm opgenomen. Als de band is teruggespoeld en dus stil staat zal de motor, die nu het slippend mechanisme moet aandrijven, veel meer stroom gaan opnemen zodat de 8C-557 open. Een eenvoudige filter stelt de gevoeligheid in en filtert de "troep", veroorzaakt door de draaiende motor uit. Ook voorziet het in enige vertraging, zodat niet elke stroompiek een output naar pen E van de USERPORT stuurt, op welk punt de CBM kan lezen of een bandspooloperatie is afgelopen.

Op pen D van de USERPORT kan de CBM nu geheel onafhankelijk zien of er een recordertoets is ingedrukt. De lijn "sense" die de CBM vertelt of een recordertoets is ingedrukt wordt onderbroken en via de dubbele transistor-inverter weer aangeslo- ten. In plaats van te worden geaard door de recorderswitch, gebeurt dat nu door het met deze schakelaar afknijpen van de eerste tor, zo- dat de tweede open. Het effect is dus voor de CBM gelijk, maar we kunnen nu zonder de schakelaar te openen de sense-line hoog maken, waardoor de motor stopt, door pen C van de USERPORT door de CBM zelf nul te laten maken.

Altereerst is het nodig de cassette recorder geschikt te maken voor ons doel. De schakeling van fig 1 past gemakkelijk op een stukje print of montaprint van 2\*4 cm.

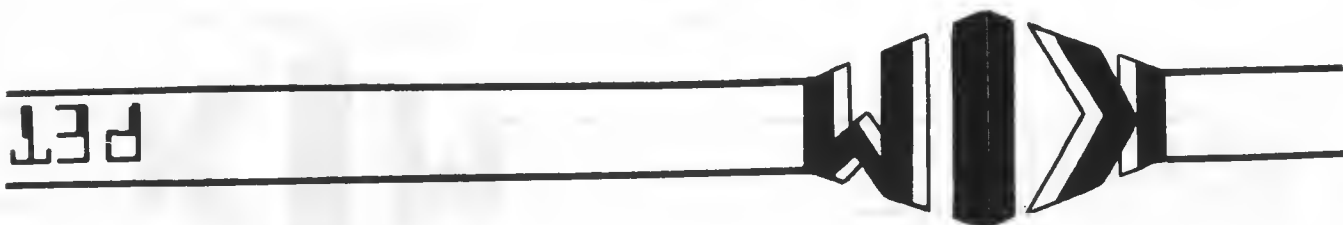
2 Een klein BASIC programma in gebruik! Zonde van de lege resterende RAM capaciteit. We zullen over meerdere programma's tegelijk in het RAM kunnen gaan beschikken. In mijn voorbeeld twee. Op dit thema mag u zelf eindeloos verder gaan hobbyen.

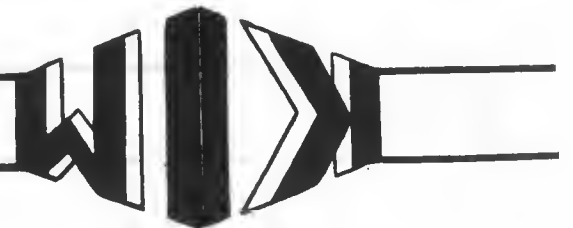
1. Het ergelijke zoeken naar een file op cassette. We rusten de PET uit met "FAST FORWARD"

Dat is de naam die ik heb gegeven aan een machinetaalprogramma van 192 bytes lang. Precies, op de byte af, passende in de buffer van de niet gebruikte tweede cassette recorder. Het programma staat bijna net zovellig als in rom, als we netjes de in de vorige aflevering besproken voorzieningen hebben aangebracht. Het programma gaat twee onhebbelijkheden van CBM/PET vermooeden:

R. Upphoff

INITIALISATOR





In het navolgende zullen we de recordermotor stop zetten door bit 0 van het DORA op 1 te zetten, zodat bit 0 van het DRA 0 wordt. We kunnen door bit 0 van het DORA weer nul te maken de motor weer vrij geven.

Laat U straks niet bedriegen door het feit dat dat bit ook bij gesloten DORA nul blijkt te zijn. De oorzaak kunt U in het schema van fig 1 zien, waar pen C laag wordt gehouden door de basis-emitter van de laatste BC 547. Het bit heeft dus "een input" die echter van geen betekenis is voor ons doel.

INPUTS (bit 0 is dus altijd 0) ADRES \$E84F

FE geen toets ingedrukt op recorder.  
FA wel toets ingedrukt op recorder.  
FB wel toets ingedrukt op recorder en einde band bereikt.  
OUTPUTS adres \$E843, 01 motor stroomloos • 00 motor kan lopen.

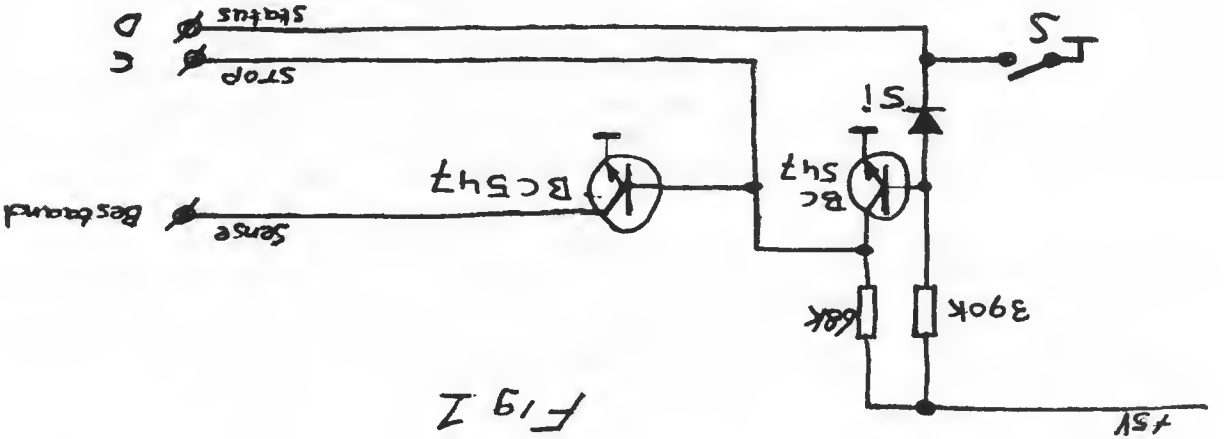
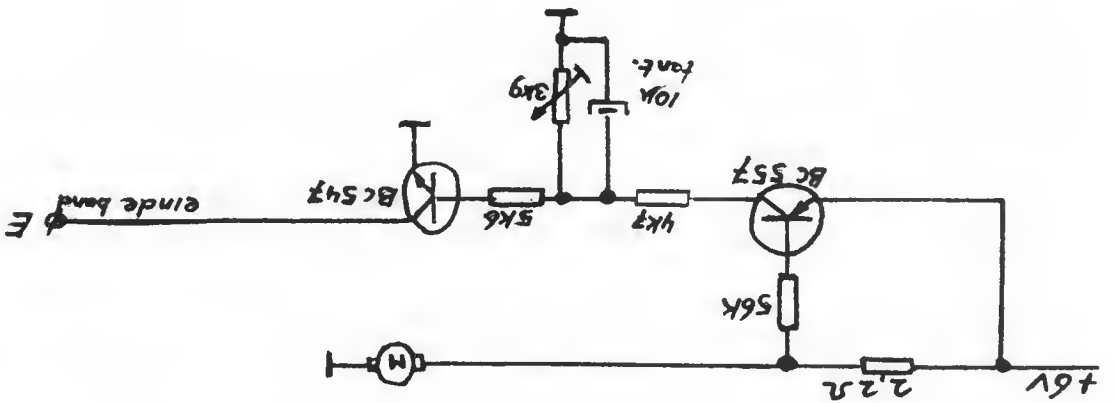


Fig 1

Een machinetaalprogramma in de 2e cassette-buffer zal straks de software gaan verzorgen. We zijn echter nog niet aan dat programma toe omdat het een "link" heeft met een tweede programma in dezelfde buffer dat ons de mogelijkheid geeft ons geheugen in twee delen van ca 4K te verdelen en dan over twee basic-programma's tegelijk te beschikken. Gelukkige bezitters van 16 of 32K machines kunnen natuurlijk nog veel verder gaan. Dan wordt het echter verplicht EPROM-programmeren want de buffer wordt dan te klein.



Wat we gaan doen is het operatiesysteem bedruiven. Dat maakt gebruik van een 24-tal locaties op de zeropage: (PET:andere adressen?) Alle adressen zijn hexadecimaal, de dataopslag is in de bekende 10-hi volgorde, zoals in elke abs instructie van de 6502.

- 28 29 Startadres basisteekst.
- 2a 2b Start variabelen opslag.
- 2c 2d Start Array variabelen.
- 2e 2f Eerste vrije RAM locatie
- 30 31 Eerste niet meer vrije RAM locatie, gebruikt voor STRING opslag
- 32 33 Eerste locatie niet meer gebruikt voor string-opslag
- 34 35 Eerste adres dat buiten het beschikbare RAM valt.

Bij een reset initialiseert de CBM als volgt:

Geheugenlocatie 0400 is het eerste RAM adres en daar komt 00 te staan als indicatie dat daar een programma gaat beginnen. Over deze 00 straks nog een leuke tip.

Nu wordt aangenomen dat BASIC aan het begin van het RAM gaat beginnen dus de pointer 28-28 wordt op 01 04 gezet. Tevens wordt hetzelfde gedaan wat het commando NEW doet, hierover zodadelijk uitleg.

Voor een 8k machine loopt het RAM tot 1FFF dus de eerste niet meer bestaande RAM-locatie is 2000 en dus zet het initialiserings-pgm. 00 20 in locatie 34 en 35.

Zodra variabelen zijn gebruikt worden de desbetreffende locaties gezet. Nu de COMMANDO's NEW en CLR

NEW maakt 30-31 en 32-33 gelijk aan 34-35, dus geen stringopslag meer aan het operatie systeem bekend.

NEW zet 00 op locaties 0401 en 0402. Dvz: een basic programma zonder inhoud dat eindigt op 0402 en dus beginnen de variabelen van dat oneigenlijke programma op 0403 en er komt dan ook op 2a en 2b, 03-04 te staan

NEW doet verder hetzelfde als CLR: wis de variabelen, dus zet alle pointers t/m 2e-2f ook op "start variabelen" dus ook 03-04.

Ik heb het bovenstaande technisch gezien niet exact beschreven maar het geeft u het juiste beeld van de pointers. Verder spreek ik over 24 bytes. Welnu 36 t/m 3f worden tijdens de programma executie voor verschillende dataopslag gebruikt. Hun inhoud interesseert ons nu niet. We moeten ze echter wel in tact laten bij de manipulaties die we ne met deze 24 bytes gaan verrichten.

Nu iets over de 00 vóór het begin van de basisteekst. Een statement wordt door 00 beëindigd en het operatiesysteem weet dan dat daarachter het nieuwe statement begint. Dat begin bestaat uit het adres waar het volgende statement zal beginnen, gevolgd door het regelnummer.

Zonder 00 dus: Geen nieuw statement: ?SYNTAX ERROR.

U kunt hiervan heerlijk misbruik maken door als u even weg moet POKE 1024,255 te typen. Runnen van het programma is dan onmogelijk en NEW vernietigt uw programma niet. Met POKE1024,0 is alles weer OK.

Door allerlei manipulaties met de besproken pointers kunt u het operatiesysteem helemaal naar uw hand zetten. Alle mogelijke kaden hier beschrijven is ondoenlijk. We zullen ons dus beperken tot de twee-deeling van het RAM. Pagina \$11 is een mooi begin voor het tweede 4k geheugen en dus zetten we daar eerst 00

Adres \$1100 moet nu ook het einde van de eerste 4K zijn en dus moet ons 8K geheugen tot 4K gemaakt.  
 Voor de tweede mogelijkheid, de laatste 4K, hebben we alle pointers weer te veranderen en dat is natuurlijk steeds een heidens karwei! Daarom creëren we een tweede 24 byte pointer set, waartoe we de laatste 24 bytes \$11E8 - 11FF van het RAM afpikken.  
 Ten slotte hebben we dan een machinetaalprogramma nodig dat beide pointer-sets kan omwisselen, al naar gelang we het eerste of het tweede programma willen gebruiken. Met deze gegevens moet U nu het hierna afgedrukte programma kunnen volgen. Enig commentaar zal daarbij wellicht gewenst zijn.

\$033a

Een subroutine die wacht op de juiste input op de USSR-PORT "geen toets ingedrukt". Door de subroutine op \$033c aan te roepen kunnen de overige inputs van de USSR PORT worden opgewacht.

\$0342

Als bij de programma start nog een recorder toets is ingedrukt zal het programma naar het nog te bespreken NMI programma gaan en alleen vragen de stop toets in te drukken. Het programma moet dan opnieuw worden gestart.  
 Vanuit basic is in direct mode met een USSR-commando naar het startadres \$0342 gesprongen. Het gehele pgm is dus een subroutine van basic. De parameter van USSR bevat het aantal 1/60 seconden dat we de band willen laten spoelen, in flo-atingpoint notatie, opgeslagen in de FAC op adres 00E5-0064 00E5 bevat daarbij de exponent.

\$0349

Indien geen recorder toets was ingedrukt neemt de routine aan dat de spoeloperatie mag beginnen en drukt de prompt -> af op het scherm. Dan wordt gewacht op het indrukken van de toets RW of FLWD op de recorder. (besproken subroutine)  
 Als de toetsdruk is uitgevoerd wordt ter bevestiging de prompt \* afgedrukt.

\$0353

\$0358

De waarde in de floating accumulator wordt naar een twee bytes integer op \$61 en 62 geconverteerd. (hi-lo)  
 Vervolgens wordt de ingebouwde klok uit de CBM afgelezen. Dit is de starttijd van het programma. We slaan hem op in de bytes \$63 en 64 van de FAC, gewoon omdat dat een even bruikbare zeeopage locatie is. De mantissa's in de FAC zijn toch verder van geen belang meer. Vervolgens wordt de twee bytes -integer (spoeittijd) van \$61-62 gehaald en met de start-tijd vermeerderd. Het resultaat komt in X en Y register, omdat dat het minste bytes kost. Dit resultaat is de stop tijd!

\$0369

\$036d

Als we niet een bepaalde tijd voortwaarts wilden spoelen, maar wilden weten hoelang het terugspoelen duurt, is het programma door ons aangeroepen met PRINT USSR(0). Nu moet dus bekeken worden of de "spoeittijd" nul is en dat kan nog steeds, ondanks alle manipulaties met de FAC, door naar de exponent op \$55 te kijken. Nemen we echter eerst aan dat daar geen 0 staat. Het wachten is nu tot de CBM klok gelijk staat aan de waarde in X en Y register. Als dat zover is de juiste tijd voorwaarts gespoeld en moet de band stil gezet gaan worden. Aan-gezien dat ook moet gebeuren naar een RWIND operatie zullen we eerst het geval USSR(0) gaan bezien. Daarna gaat het pgm in alle gevallen weer dezelfde weg.

\$0377

USR was 0. Nu wordt gewacht op het einde van de rewind-operatie, waarbij zich een klein probleem voor doet: De super-snelle processor is al met dit programma deel bezig als de aanloopstroom van de cassette-motor nog hoog is. Dit zou nu als "einde band" worden opgevat en daarom wordt ca 2 sec. gewacht. Pas daarna (weer de subroutine) wordt op ontvangst van het einde-band signaal gewacht.

\$384

Of het einde van de FWD-spoeltijd of eindband nu de reden is doet er niet toe. De motor moet worden gestopt. Eerst wordt hier echter berekend hoe lang de spoel-operatie heeft geduurd. Bij RW is dat het DOEL van het pgm. Bij FWD een leuke controle. Het resultaat van de berekening komt in Y en Accumulator en een subroutine uit de basic-interpretator plaatst het resultaat netjes in floatingspoint in de FAC.

\$390

Gevraagd wordt de stop-toets in te drukken van de recorder en: De motor wordt stil gezet. De enige subroutine laat weer wachten op het indrukken van de stop-toets (=geen toets gedrukt) De motor wordt dan weer vrijgegeven en een RTS keert terug naar basic waar de inhoud van de FAC, de spoeltijd wordt afgedrukt.

\$0395

\$039d

Om USR te kunnen gebruiken staat in het operatiesysteem op adres \$ 0000 een constante 4C JMP. Daarnaast moet de gebruiker zelf het juiste adres invullen. Uit pure luiheid laten we dat echter een routine doen op \$03a3, door SYS 931 te typen. Dat hoeft maar één keer, na het laden van het programma te gebeuren en natuurlijk na gebruik van uw zelf aangebrachte reseschkelaar. De routine verandert ook de NWI-vector op \$0094 en 0095, teneinde NWI via ons eigen interrupt-programma naar het normale interrupt-programma op \$C3B9 te sturen. PET-bezitters moeten dit stukje pgm helaas weglaten omdat hun machine niet over een NWI-RAM-vector beschikt. Het NWI-pgm op 03b2 doet niets anders dan eerst even kijken of er geen cassette I/O operatie bezig is, door het LSB van de RW-vector te controleren. Zoja wordt NWI haastig met een RTI de laan uit gestuurd. Voorts controleert de routine, die we immers veel zullen gebruiken op ons FWD-programma te onderbreken, of we netjes de stop-toets van de recorder hebben ingedrukt. De motor wordt ook terloops even gestopt. Bij elk NWI-gebruik wordt ons tenslotte nog even verteld welk start-adres ons RAM-programma heeft, waarnaover meer in dat deel van het tweede programma in de buffer. We zullen nu eerst de gebruiksaanwijzing behandelen, van de FWD-faciliteit.

Als u een programma naar cassette gaat schrijven bepaald u en niets of niemand anders voortaan waar op de cassette dat programma komt. Udoot dat als volgt:

PRINTUSR(3740):SAVE"NAAM"

\* 3740

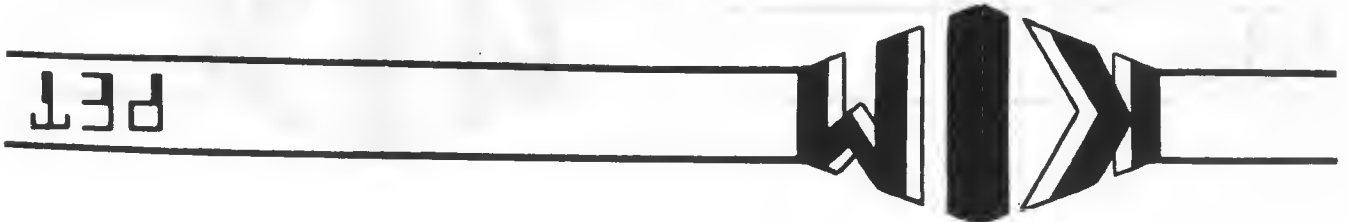
PRESS PLAY & RECORD ON TAPE 1

OK

WRITING NAAM

READY.





Om het programma later terug te vinden typed U:

PRINTUSR(3740):LOAD

de naam mag u gerust weg laten want uw CBM vindt daar al een het goede programma. Er blijft nu nog een probleem: waar mag u het volgen de programma wegschrijven!

PRINTUSR(0)

\* 1731

READY.

Zoek nu in de grafiek van fig. 2 de factor  $V$  op en vraag:

Λ/ΤΣΛΤΐ

Notteer het antwoord als plaats voor de volgende filie.

U kunt voor elk type cassette het snijpunt vinden door de totale speeltijd van de cassette te weten. De grafiek is wat onnauwkeurig omdat alierhande invloeden zoals langzamer lopende motor aan het einde van RW niet zijn verwerkt. Deze fouten moet men laten voor wat ze zijn want ze vormen een klein deel van de gewenste verlichtingswaarde.

Het tweede deel van het programma geeft u de mogelijkheid met SYSG59 van RAM te verwijzen: onderste of bovenste 4K. Het programma moet na wat ik over de pointers verteld heb voor zich spreken. Vergeet niet na de eerste SYSG59 het commando NEW te typen. Een volgende keer zal ik desgevraagd nog een aantal mogelijkheden met deze pointers onder de loep nemen, die te maken hebben met programma OVERLAY en machinetaalroutines in het RAM naast het bijbehorend basic. Voorts is nog een SOFTWARE blokgeeft generator in de maak met zeer grote frequentienauwkeurigheid.

Voorts is nog een SOFTWARE blokje generator in de maak met zeer grote frequentienauwkeurigheid.

PROGRAMMA

```
033a a9 fe lda imm          st cont. controller status recorder
033c cd 4f e8 cmp abs          wach op input user port
033f d0 fb bne rel          bne 033c
0341 60                                bne rel          bne 033c
```

```
0342 a9 fe lda imm cont.stopopts
0343 cd 4f cmp abs
0344 d0 70 bne rel bne 03b9 break pgm af.
0345 a9 5f lda imm print prompt →
```

035b	as	8f	lda	zpg	lsb clock
035f	85	64	sta	zpg	sta lsb clock startt:jd
0361	65	62	adc	zpg	lsb stop:jd
0362	88		tay	tmp	sta in y lsb stop:jd
0367	85	62	lda	zpg	lda qsw als bover loop qsw

```

036b f0 0a beq rel indien 0 rew,zoniet ffw.
036d e4 8e cpx zpg wacht op einde spoeltijd.
036f d0 fc bne rel bne 036d
0371 c4 8f cpy zpg
0373 d0 fc bne rel bne 0371
0375 f0 0d beq rel beq 0384
0377 a5 64 lda zpg wacht op einde bandsignaal
0379 69 20 adc lmm wacht op einde aanloopstroom motor
037b c5 8f cmp zpg bne rel bne 037b
037f a9 f8 lda zpg zpg
0381 20 3c jsr abs jsr cont. wacht op einde bandsignaal.
0384 a5 8f lda zpg bereken looptijd band
0386 e5 64 sbc zpg
0388 a8 tay lmp
0389 a5 8e lda zpg
038b e5 63 sbc zpg
038d 20 6d jsr abs jsr int-flp (basic int.)
0390 a9 a6 lda lmm print prompt
0392 20 32 jsr abs jsr wrt (basic int.)
0395 a9 01 lda lmm blokkeer motor (stop)
0397 8d 43 sta abs jsr cont. wacht op stopstoets
039d a9 00 lda lmm geef motor vrij
03a2 60 rts lmp Terug naar basic/of einde srt.
03a3 a9 42 lda lmm initialiseer NMI en USR (SYS 931)
03a5 85 01 sta zpg adres usr
03a7 a9 03 lda lmm
03a9 85 02 sta zpg
03ab 85 95 sta zpg page usr (vector 0001/0002 geseet)
03ad a9 b2 lda lmm
03af 85 94 sta zpg adres nmi (vector 0094/0095 geseet)
03b1 60 rts lmp Terug naar basic
03b2 a9 2e lda lmm "masked" nmi programma
03b4 c5 90 cmp zpg compare irq-vector: leesoperatie gaande ?
03b6 f0 01 beq rel beq 0309
03b8 40 rti lmp leesoperatie: "ignore" NMI
03b9 20 90 jsr abs greek lopend spoelprogramma/ingedrukte toets-af!
03bc 4c f2 jmp abs print geheugen(RAM) status
03bf ad 00 lda abs memory control program.: Controleer status.
03c2 f0 1b beq rel beq 03df Initialiseer niet onnodig.(fout)
03c4 a9 00 lda lmm initialiseer geheugendeeling
03c6 8d 00 sta abs 00 pointer voor tweede RAM helft
03c9 a9 01 lda lmm creeer tweede pointer serie,
03cb 8d e8 sta abs en:
03ce a9 1f lda lmm wijzig datapointers.
03d0 85 35 sta zpg
03d2 8d e9 sta abs
03d5 a9 e8 lda lmm
03d7 8d f4 sta abs
03da a9 1f lda lmm
03dc 8d f5 sta abs

```

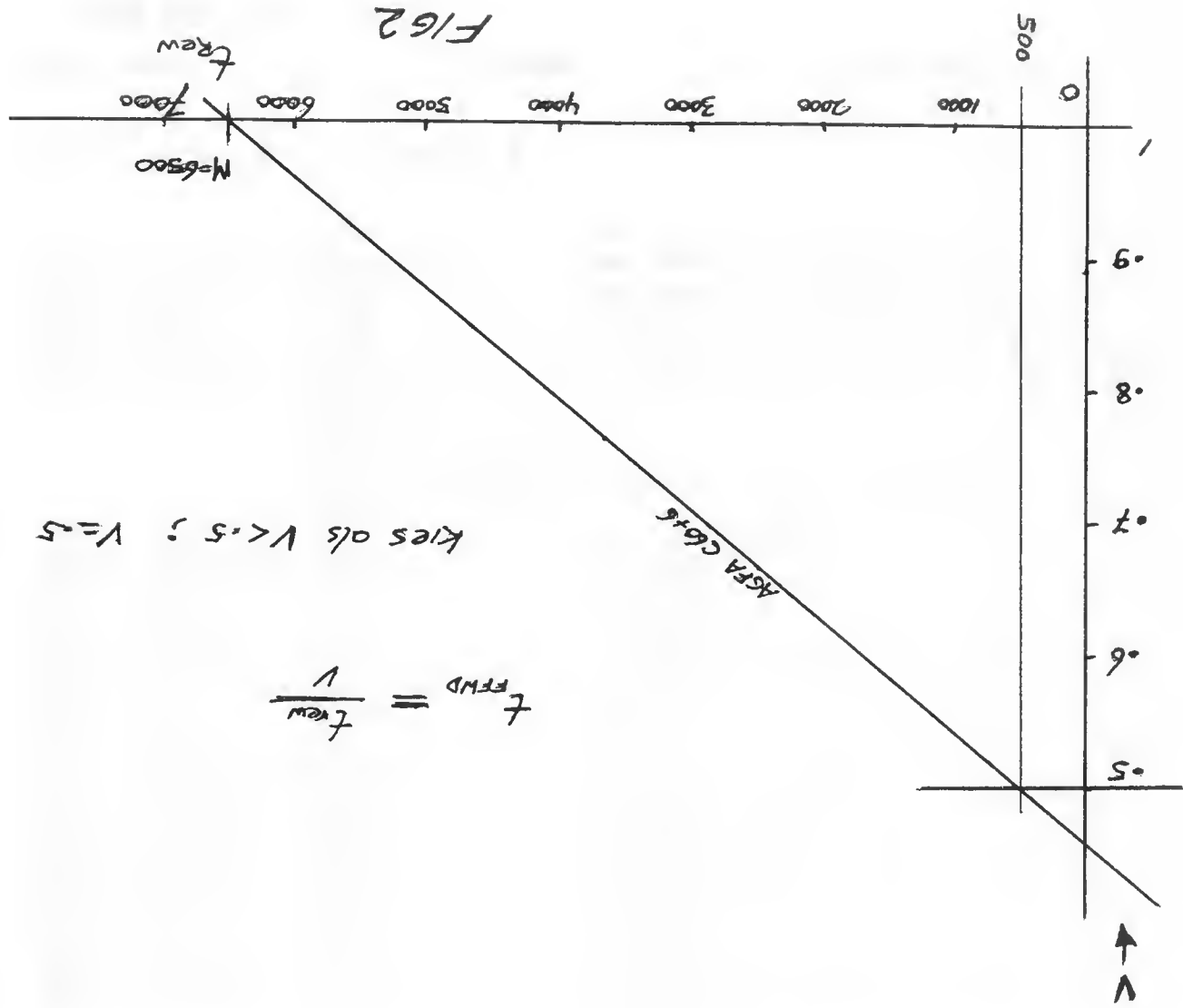


PET

```

03df a2 00 ldx lmm wissel gehen gebuik (sys 959)
03e1 b5 28 ldx zpx
03e3 48 pha lmp
03e4 bd e8 1f ldx abx
03e7 95 28 sta zpx
03e9 68 pla lmp
03ea 9d e8 1f sta abx
03ed e8 lnx lmp
03ee e0 19 cpx lmm
03f0 d0 ef bne rel
03f2 a5 29 ldx zpg
03f4 20 75 jsr abs
03f7 4c 89 c3 jmp abs
warm start basic.
print gehen keuze(status)
jsr wrob (basic int.) "type a byte"

```





```
0010 ;**** RAE PRINTER BOOTSTRAP ****
0020 ;
0030 ;Laden via normale SYM L2 routine,
0040 ;vervoers G 0 indrukken, waarna men
0050 ;direct in de koude start van de RAE
0060 ;uitkomt.
0080 ;
0090 ;
0100 ;BA 0000
0110 ;OS
0120 JSR $B8B6 acces
0130 LDA #$0A set in/out registers
0140 STA $A80C
0150 LDA #$FF
0160 STA $A803 dataadr. reg. via #2
0170 LDA #$0A
0180 STA $AC03 dataadr. reg. via #3
0190 LDX #$0E move patch
0200 MOVE
0210 STA $A5FF, X scbutf -1
0220 DEX
0230 BNE MOVE
0240 LDA #$00 verander outvector
0250 STA $A6E4
0260 LDA #$A6
0270 STA $A6E5
0280 JSR $B89C naces
0290 JMP $B800 *RAE*
0300 ;
0310 ; EIGENLIJKE PATCH (Wordt in SCBUTF
0320 ;geplaatst.)
0330 ;
0340 ;
0350 KJK
0360 BVC TERUG geen select, aa terug
0370 BMI KJK wacht tot busy weg is
0380 STA $A801 verzend karakter
0390 TERUG
0400 ;
0410 ;
0420 ;
0430 ;
0440 ;
0450 ;
0460 ;
0470 ;
0480 ;
0490 ;
0500 ;
0510 ;
0520 ;
0530 ;
0540 ;
0550 ;
0560 ;
0570 ;
0580 ;
0590 ;
0600 ;
0610 ;
0620 ;
0630 ;
0640 ;
0650 ;
0660 ;
0670 ;
0680 ;
0690 ;
0700 ;
0710 ;
0720 ;
0730 ;
0740 ;
0750 ;
0760 ;
0770 ;
0780 ;
0790 ;
0800 ;
0810 ;
0820 ;
0830 ;
0840 ;
0850 ;
0860 ;
0870 ;
0880 ;
0890 ;
0900 ;
0910 ;
0920 ;
0930 ;
0940 ;
0950 ;
0960 ;
0970 ;
0980 ;
0990 ;
1000 ;
1010 ;
1020 ;
1030 ;
1040 ;
1050 ;
1060 ;
1070 ;
1080 ;
1090 ;
1100 ;
1110 ;
1120 ;
1130 ;
1140 ;
1150 ;
1160 ;
1170 ;
1180 ;
1190 ;
1200 ;
1210 ;
1220 ;
1230 ;
1240 ;
1250 ;
1260 ;
1270 ;
1280 ;
1290 ;
1300 ;
1310 ;
1320 ;
1330 ;
1340 ;
1350 ;
1360 ;
1370 ;
1380 ;
1390 ;
1400 ;
1410 ;
1420 ;
1430 ;
1440 ;
1450 ;
1460 ;
1470 ;
1480 ;
1490 ;
1500 ;
1510 ;
1520 ;
1530 ;
1540 ;
1550 ;
1560 ;
1570 ;
1580 ;
1590 ;
1600 ;
1610 ;
1620 ;
1630 ;
1640 ;
1650 ;
1660 ;
1670 ;
1680 ;
1690 ;
1700 ;
1710 ;
1720 ;
1730 ;
1740 ;
1750 ;
1760 ;
1770 ;
1780 ;
1790 ;
1800 ;
1810 ;
1820 ;
1830 ;
1840 ;
1850 ;
1860 ;
1870 ;
1880 ;
1890 ;
1900 ;
1910 ;
1920 ;
1930 ;
1940 ;
1950 ;
1960 ;
1970 ;
1980 ;
1990 ;
2000 ;
2010 ;
2020 ;
2030 ;
2040 ;
2050 ;
2060 ;
2070 ;
2080 ;
2090 ;
2100 ;
2110 ;
2120 ;
2130 ;
2140 ;
2150 ;
2160 ;
2170 ;
2180 ;
2190 ;
2200 ;
2210 ;
2220 ;
2230 ;
2240 ;
2250 ;
2260 ;
2270 ;
2280 ;
2290 ;
2300 ;
2310 ;
2320 ;
2330 ;
2340 ;
2350 ;
2360 ;
2370 ;
2380 ;
2390 ;
2400 ;
2410 ;
2420 ;
2430 ;
2440 ;
2450 ;
2460 ;
2470 ;
2480 ;
2490 ;
2500 ;
2510 ;
2520 ;
2530 ;
2540 ;
2550 ;
2560 ;
2570 ;
2580 ;
2590 ;
2600 ;
2610 ;
2620 ;
2630 ;
2640 ;
2650 ;
2660 ;
2670 ;
2680 ;
2690 ;
2700 ;
2710 ;
2720 ;
2730 ;
2740 ;
2750 ;
2760 ;
2770 ;
2780 ;
2790 ;
2800 ;
2810 ;
2820 ;
2830 ;
2840 ;
2850 ;
2860 ;
2870 ;
2880 ;
2890 ;
2900 ;
2910 ;
2920 ;
2930 ;
2940 ;
2950 ;
2960 ;
2970 ;
2980 ;
2990 ;
3000 ;
3010 ;
3020 ;
3030 ;
3040 ;
3050 ;
3060 ;
3070 ;
3080 ;
3090 ;
3100 ;
3110 ;
3120 ;
3130 ;
3140 ;
3150 ;
3160 ;
3170 ;
3180 ;
3190 ;
3200 ;
3210 ;
3220 ;
3230 ;
3240 ;
3250 ;
3260 ;
3270 ;
3280 ;
3290 ;
3300 ;
3310 ;
3320 ;
3330 ;
3340 ;
3350 ;
3360 ;
3370 ;
3380 ;
3390 ;
3400 ;
3410 ;
3420 ;
3430 ;
3440 ;
3450 ;
3460 ;
3470 ;
3480 ;
3490 ;
3500 ;
3510 ;
3520 ;
3530 ;
3540 ;
3550 ;
3560 ;
3570 ;
3580 ;
3590 ;
3600 ;
3610 ;
3620 ;
3630 ;
3640 ;
3650 ;
3660 ;
3670 ;
3680 ;
3690 ;
3700 ;
3710 ;
3720 ;
3730 ;
3740 ;
3750 ;
3760 ;
3770 ;
3780 ;
3790 ;
3800 ;
3810 ;
3820 ;
3830 ;
3840 ;
3850 ;
3860 ;
3870 ;
3880 ;
3890 ;
3900 ;
3910 ;
3920 ;
3930 ;
3940 ;
3950 ;
3960 ;
3970 ;
3980 ;
3990 ;
4000 ;
4010 ;
4020 ;
4030 ;
4040 ;
4050 ;
4060 ;
4070 ;
4080 ;
4090 ;
4100 ;
4110 ;
4120 ;
4130 ;
4140 ;
4150 ;
4160 ;
4170 ;
4180 ;
4190 ;
4200 ;
4210 ;
4220 ;
4230 ;
4240 ;
4250 ;
4260 ;
4270 ;
4280 ;
4290 ;
4300 ;
4310 ;
4320 ;
4330 ;
4340 ;
4350 ;
4360 ;
4370 ;
4380 ;
4390 ;
4400 ;
4410 ;
4420 ;
4430 ;
4440 ;
4450 ;
4460 ;
4470 ;
4480 ;
4490 ;
4500 ;
4510 ;
4520 ;
4530 ;
4540 ;
4550 ;
4560 ;
4570 ;
4580 ;
4590 ;
4600 ;
4610 ;
4620 ;
4630 ;
4640 ;
4650 ;
4660 ;
4670 ;
4680 ;
4690 ;
4700 ;
4710 ;
4720 ;
4730 ;
4740 ;
4750 ;
4760 ;
4770 ;
4780 ;
4790 ;
4800 ;
4810 ;
4820 ;
4830 ;
4840 ;
4850 ;
4860 ;
4870 ;
4880 ;
4890 ;
4900 ;
4910 ;
4920 ;
4930 ;
4940 ;
4950 ;
4960 ;
4970 ;
4980 ;
4990 ;
5000 ;
5010 ;
5020 ;
5030 ;
5040 ;
5050 ;
5060 ;
5070 ;
5080 ;
5090 ;
5100 ;
5110 ;
5120 ;
5130 ;
5140 ;
5150 ;
5160 ;
5170 ;
5180 ;
5190 ;
5200 ;
5210 ;
5220 ;
5230 ;
5240 ;
5250 ;
5260 ;
5270 ;
5280 ;
5290 ;
5300 ;
5310 ;
5320 ;
5330 ;
5340 ;
5350 ;
5360 ;
5370 ;
5380 ;
5390 ;
5400 ;
5410 ;
5420 ;
5430 ;
5440 ;
5450 ;
5460 ;
5470 ;
5480 ;
5490 ;
5500 ;
5510 ;
5520 ;
5530 ;
5540 ;
5550 ;
5560 ;
5570 ;
5580 ;
5590 ;
5600 ;
5610 ;
5620 ;
5630 ;
5640 ;
5650 ;
5660 ;
5670 ;
5680 ;
5690 ;
5700 ;
5710 ;
5720 ;
5730 ;
5740 ;
5750 ;
5760 ;
5770 ;
5780 ;
5790 ;
5800 ;
5810 ;
5820 ;
5830 ;
5840 ;
5850 ;
5860 ;
5870 ;
5880 ;
5890 ;
5900 ;
5910 ;
5920 ;
5930 ;
5940 ;
5950 ;
5960 ;
5970 ;
5980 ;
5990 ;
6000 ;
6010 ;
6020 ;
6030 ;
6040 ;
6050 ;
6060 ;
6070 ;
6080 ;
6090 ;
6100 ;
6110 ;
6120 ;
6130 ;
6140 ;
6150 ;
6160 ;
6170 ;
6180 ;
6190 ;
6200 ;
6210 ;
6220 ;
6230 ;
6240 ;
6250 ;
6260 ;
6270 ;
6280 ;
6290 ;
6300 ;
6310 ;
6320 ;
6330 ;
6340 ;
6350 ;
6360 ;
6370 ;
6380 ;
6390 ;
6400 ;
6410 ;
6420 ;
6430 ;
6440 ;
6450 ;
6460 ;
6470 ;
6480 ;
6490 ;
6500 ;
6510 ;
6520 ;
6530 ;
6540 ;
6550 ;
6560 ;
6570 ;
6580 ;
6590 ;
6600 ;
6610 ;
6620 ;
6630 ;
6640 ;
6650 ;
6660 ;
6670 ;
6680 ;
6690 ;
6700 ;
6710 ;
6720 ;
6730 ;
6740 ;
6750 ;
6760 ;
6770 ;
6780 ;
6790 ;
6800 ;
6810 ;
6820 ;
6830 ;
6840 ;
6850 ;
6860 ;
6870 ;
6880 ;
6890 ;
6900 ;
6910 ;
6920 ;
6930 ;
6940 ;
6950 ;
6960 ;
6970 ;
6980 ;
6990 ;
7000 ;
7010 ;
7020 ;
7030 ;
7040 ;
7050 ;
7060 ;
7070 ;
7080 ;
7090 ;
7100 ;
7110 ;
7120 ;
7130 ;
7140 ;
7150 ;
7160 ;
7170 ;
7180 ;
7190 ;
7200 ;
7210 ;
7220 ;
7230 ;
7240 ;
7250 ;
7260 ;
7270 ;
7280 ;
7290 ;
7300 ;
7310 ;
7320 ;
7330 ;
7340 ;
7350 ;
7360 ;
7370 ;
7380 ;
7390 ;
7400 ;
7410 ;
7420 ;
7430 ;
7440 ;
7450 ;
7460 ;
7470 ;
7480 ;
7490 ;
7500 ;
7510 ;
7520 ;
7530 ;
7540 ;
7550 ;
7560 ;
7570 ;
7580 ;
7590 ;
7600 ;
7610 ;
7620 ;
7630 ;
7640 ;
7650 ;
7660 ;
7670 ;
7680 ;
7690 ;
7700 ;
7710 ;
7720 ;
7730 ;
7740 ;
7750 ;
7760 ;
7770 ;
7780 ;
7790 ;
7800 ;
7810 ;
7820 ;
7830 ;
7840 ;
7850 ;
7860 ;
7870 ;
7880 ;
7890 ;
7900 ;
7910 ;
7920 ;
7930 ;
7940 ;
7950 ;
7960 ;
7970 ;
7980 ;
7990 ;
8000 ;
8010 ;
8020 ;
8030 ;
8040 ;
8050 ;
8060 ;
8070 ;
8080 ;
8090 ;
8100 ;
8110 ;
8120 ;
8130 ;
8140 ;
8150 ;
8160 ;
8170 ;
8180 ;
8190 ;
8200 ;
8210 ;
8220 ;
8230 ;
8240 ;
8250 ;
8260 ;
8270 ;
8280 ;
8290 ;
8300 ;
8310 ;
8320 ;
8330 ;
8340 ;
8350 ;
8360 ;
8370 ;
8380 ;
8390 ;
8400 ;
8410 ;
8420 ;
8430 ;
8440 ;
8450 ;
8460 ;
8470 ;
8480 ;
8490 ;
8500 ;
8510 ;
8520 ;
8530 ;
8540 ;
8550 ;
8560 ;
8570 ;
8580 ;
8590 ;
8600 ;
8610 ;
8620 ;
8630 ;
8640 ;
8650 ;
8660 ;
8670 ;
8680 ;
8690 ;
8700 ;
8710 ;
8720 ;
8730 ;
8740 ;
8750 ;
8760 ;
8770 ;
8780 ;
8790 ;
8800 ;
8810 ;
8820 ;
8830 ;
8840 ;
8850 ;
8860 ;
8870 ;
8880 ;
8890 ;
8900 ;
8910 ;
8920 ;
8930 ;
8940 ;
8950 ;
8960 ;
8970 ;
8980 ;
8990 ;
9000 ;
9010 ;
9020 ;
9030 ;
9040 ;
9050 ;
9060 ;
9070 ;
9080 ;
9090 ;
9100 ;
9110 ;
9120 ;
9130 ;
9140 ;
9150 ;
9160 ;
9170 ;
9180 ;
9190 ;
9200 ;
9210 ;
9220 ;
9230 ;
9240 ;
9250 ;
9260 ;
9270 ;
9280 ;
9290 ;
9300 ;
9310 ;
9320 ;
9330 ;
9340 ;
9350 ;
9360 ;
9370 ;
9380 ;
9390 ;
9400 ;
9410 ;
9420 ;
9430 ;
9440 ;
9450 ;
9460 ;
9470 ;
9480 ;
9490 ;
9500 ;
9510 ;
9520 ;
9530 ;
9540 ;
9550 ;
9560 ;
9570 ;
9580 ;
9590 ;
9600 ;
9610 ;
9620 ;
9630 ;
9640 ;
9650 ;
9660 ;
9670 ;
9680 ;
9690 ;
9700 ;
9710 ;
9720 ;
9730 ;
9740 ;
9750 ;
9760 ;
9770 ;
9780 ;
9790 ;
9800 ;
9810 ;
9820 ;
9830 ;
9840 ;
9850 ;
9860 ;
9870 ;
9880 ;
9890 ;
9900 ;
9910 ;
9920 ;
9930 ;
9940 ;
9950 ;
9960 ;
9970 ;
9980 ;
9990 ;
10000 ;
```

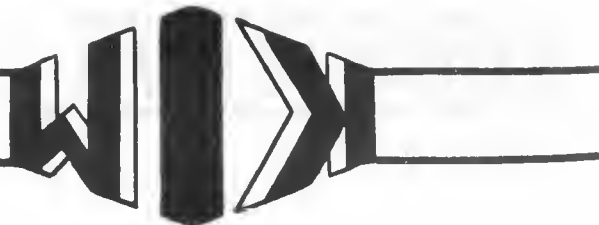
LABEL FILE: [ / = EXTERNAL ]

MOVE=0014

KJK=002D

TERUG=0037

//0000,003A,003A



K I M S C H A A K P R O G R A M M A

- Ervaringen
- Informatie-uitwisseling
- Verdere ontwikkelingen

Na de publicatie van het KIM-schaakprogramma in de KIM-Kenner nr. 11 zijn er een aantal leuke reacties gekomen. Deze reacties hebben geleid tot een aantal vragen over het programma, en een aantal uitbreidingen op het programma. Deze uitbreidingen zijn tot stand gekomen in samenwerking met Joop Tervooren, die veel van het testwerk voor zijn rekening heeft genomen, en die o.m. een vrij uitgebreide tabel met openingen voor het KIM-schaakprogramma heeft opgesteld.

1. Het schaakprogramma is opgenomen in de KIM-club programma-bibliotheek, zowel in de versie KIM speelt wit als in de versie KIM speelt zwart. Belangstellenden kunnen tijdens de KIM-club bijeenkomsten een copie maken van deze programma's. (Zie regels voor het copieren uit de programma-bibliotheek.)

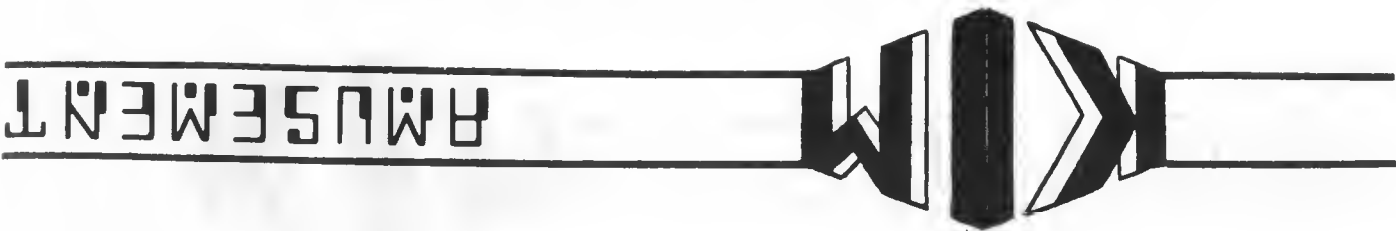
2. Er is enige onduidelijkheid over welke aanpassingen moeten worden gemaakt om de KIM zwart te laten spelen. Voor alle duidelijkheid volgt hier een opsomming van de wijzigingen ten opzichte van het programma: in de versie KIM speelt wit:

STY	DSP1	adres	00 10	84 F9
JMP	CODE	"	00 12	4D D4 17
CPY	X'3B'	"	00 C4	CO 3B

Deze wijzigingen moeten worden aangebracht in deel 1 van het programma. (Dus vóórdat deel twee is ingelezen). Hierna kan het gewijzigde deel 1 weer op de band worden gezet (adres 0000-0400); deel twee moet daar ongewijzigd achter worden geplaatst. Wanneer men de wijzigingen vlak voor het spelen met de hand wil inbrengen kan dit nadat beide delen zijn ingelezen. (Maar vóórdat de tweede keer op GO wordt gedrukt.) In dat geval wel eerst de accumulator op nul zetten, (00F3 00), voordat gestart wordt op 0000.

3. Op de volgende pagina's vindt u een openingen-routine en een tabel met openingen voor het KIM-schaakprogramma. Dit is uitgewerkt en getest door Joop Tervooren. (Er is wel meer dan 1 k geheugen voor nodig.)

4. Op de daaropvolgende pagina een stukje programma waarmee het schaakprogramma snel kan worden geïnstalleerd. Door dit stukje programma in te lezen, nadat een partij is gespeeld, kan snel met een nieuwe partij worden begonnen. (Ontwikkeld door Joop Tervooren)



5. Op de pagina daaropvolgend een routine die het schaakprogramma versnelt. Ook hier is meer geheugenruimte voor nodig. Deze routine is nog niet geheel uitgetest. Wie probeert dit verder uit ?
6. Op - en aanmerkingen graag aan Theo Kortekaas tel. 02977-21888

DATA /Adressen velden KIM-schaakprogramma T.K.K.

8	4e	4f	50	51	52	53	54	55
7	46	47	48	49	4a	4b	4c	4d
6	3e	3f	40	41	42	43	44	45
5	36	37	38	39	3a	3b	3c	3d
4	2e	2f	30	31	32	33	34	35
3	26	27	28	29	2a	2b	2c	2d
2	1e	1f	20	21	22	23	24	25
1	16	17	18	19	1a	1b	1c	1d

Veld-aanduiding  
schaakbord.

A	B	C	D	E	F	G	H
84	83	85	86	82	85	83	84
80	80	80	80	80	80	80	80
00	=	leeg	veld::				

stukken-code  
geplaatst op  
schaakbord::

A	B	C	D	E	F	G	H
01	01	01	01	01	01	01	01
C4	C3	C5	C6	C2	C5	C3	C4

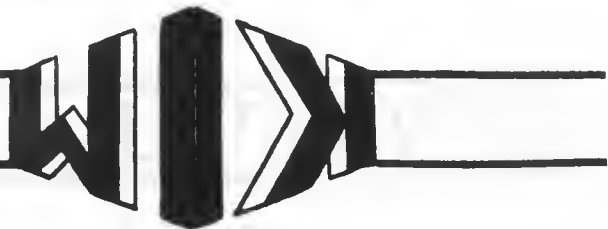
Stukken opzetten tijdens het spel::: Problemen??

RS-AD-veldadres - DA-stukcode - AD - en op beschreven  
manier zet invoeren.

B2-B4, word dan OC, in tab1: en IC in tab2: in tab3 komt dan de adressaanduiding, van tab1, waar de volgende zet staat aangegeven, in "LOW Order" aanduiden. OC dus voor 02, wil men alleen deze zet mogelijk maken, dan ipv "OC", 4C aanduiden; In het eerste geval kan de KIM kiezen uit meerdere opeenvolgende zetten, zulke reeksen afsluiten met die ipv., of +4 links in de code.

20 31 05

00(KIM speelt met zwart, dan 01)



Om KIM met zwart te laten spelen moet 66k worden gewijzigd  
0027 EA in 0027 00, alvorens te initiëren

Øtab1 Øtab2 Øtab3  
(0700) Dit alles kan overal elders in het geheugen,  
(0800) mits aanpassingen voor de plaatsbepaling.  
(0900)

zetten adres TAB1 TAB2 TAB3

zetten adres TAB1 TAB2 TAB3

b4xc3	2D	2E	73	59	12	30
d7-d5	2F	73	23	73	23	31
c6-a5	2F	5B	23	6A	20	32
d4-d5	30	5C	23	5C	23	33
e4xd5	31	5A	13	5A	13	34
c4-d3	32	52	2D	52	2D	35
c3-f6	33	6D	23	6D	23	36
f6xd5	34	73	23	73	23	37
d7-d5	35	73	23	73	23	38
f1-e1	36	45	04	45	04	39
f3xe5	37	55	24	55	24	3A
e4-e5	38	5C	24	5C	24	3B
c6-e7	39	6A	34	6A	34	3C

e2-e4	1	0C	05	01
d2-d4	2	0B	1B	60
c2-c4	3	0A	1A	98
b2-b4	4	49	19	D1
e7-e5	5	34	24	08
c7-c5	6	32	22	3C
e7-e6	7	74	2C	4e
01-f3	8	06	15	0A
d2-d4	9	4B	1B	0B
b8-c6	A	79	2A	0C
e5xd4	B	64	1B	0E
f1-c4	C	05	1A	0F
f1-b5	D	45	21	10
d1-d4	E	43	1B	11
f8-c5	F	7D	22	12
a7-a6	10	70	28	13
b8-c6	11	79	2A	14
c2-c3	12	4A	12	15
b5-a4	13	61	18	16
d4-e3	14	5B	14	17
08-f6	15	7E	2D	18
08-f6	16	7E	2D	19
08-f6	17	7E	2D	1A
d2-d4	18	4B	1B	1B
e1-01	19	44	06	1C
b1-c3	1A	41	12	1D
e5xd4	1B	64	1B	1E
f8-e7	1C	7D	34	1F
f8-b4	1D	7D	19	20
c3xd4	1E	52	1B	21
f1-e1	1F	45	04	22
c1-d2	20	42	0B	23
c5-b4	21	62	19	24
b7-b5	22	71	21	25
e8-08	23	7C	3E	26
b1-c3	24	41	12	27
a4-b3	25	58	11	28
e1-c1	26	44	02	29
f6xe4	27	64	1C	2A
e8-08	28	7C	3E	2B
f8-e8	29	7D	3C	2C
e1-01	2A	44	06	2D
c2-c3	2B	4A	12	2E
f1-c4	2C	45	1A	2F

Code label / opheningen

8	38	39	3A	3B	3C	3D	3E	3F
7	30	31	32	33	34	35	36	37
6	28	29	2A	2B	2C	2D	2E	2F
5	20	21	22	23	24	25	26	27
4	18	19	1A	1B	1C	1D	1E	1F
3	10	11	12	13	14	15	16	17
2	08	09	0A	0B	0C	0D	0E	0F
1	00	01	02	03	04	05	06	07

A B C D E F G H

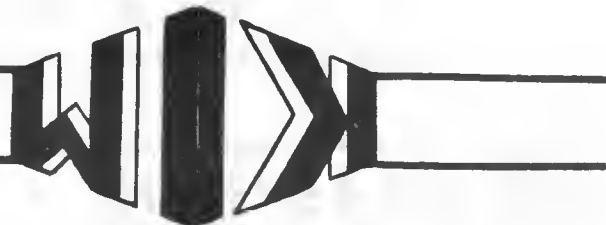
0 1





zetteon adres TAB1 TAB2 TAB3

71	45	70
72	73	71
73	46	72
74	22	73
75	06	74
76	1A	75
77	1A	76
78	33	77
79	00	78
7A	29	79
7B	23	7A
7C	59	7B
7D	15	7C
7E	22	7D
7F	23	7E
80	1B	7F
81	1B	80
82	23	81
83	1C	82
84	2A	83
85	21	84
86	1C	85
87	06	86
88	2D	87
89	26	88
90	34	89
91	13	90
92	2C	91
93	25	92
94	6C	93
95	43	94
96	23	95
97	6E	96
98	4C	97
99	79	98
9A	71	99
9B	58	9A
9C	59	9B
9D	46	9C
9E	15	9D
9F	72	9E
00	5A	9F
01	58	00
02	59	01
03	46	02
04	15	03
05	22	04
06	23	05
07	1B	06
08	1B	07
09	23	08
0A	5A	09
0B	58	0A
0C	59	0B
0D	46	0C
0E	15	0D
0F	72	0E
10	71	0F
11	79	10
12	7A	11
13	7B	12
14	7C	13
15	7D	14
16	7E	15
17	7F	16
18	80	17
19	81	18
1A	82	19
1B	83	1A
1C	84	1B
1D	85	1C
1E	86	1D
1F	87	1E
20	88	1F
21	89	20
22	90	21
23	91	22
24	92	23
25	93	24
26	94	25
27	95	26
28	96	27
29	97	28
2A	98	29
2B	99	2A
2C	00	2B
2D	01	2C
2E	02	2D
2F	03	2E
30	04	2F
31	05	30
32	06	31
33	07	32
34	08	33
35	09	34
36	0A	35
37	0B	36
38	0C	37
39	0D	38
3A	0E	39
3B	0F	3A
3C	10	3B
3D	11	3C
3E	12	3D
3F	13	3E
40	14	3F
41	15	40
42	16	41
43	17	42
44	18	43
45	19	44
46	1A	45
47	1B	46
48	1C	47
49	1D	48
4A	1E	49
4B	1F	4A
4C	20	4B
4D	21	4C
4E	22	4D
4F	23	4E
50	24	4F
51	25	50
52	26	51
53	27	52
54	28	53
55	29	54
56	2A	55
57	2B	56
58	2C	57
59	2D	58
5A	2E	59
5B	2F	5A
5C	30	5B
5D	31	5C
5E	32	5D
5F	33	5E
60	34	5F
61	35	60
62	36	61
63	37	62
64	38	63
65	39	64
66	3A	65
67	3B	66
68	3C	67
69	3D	68
6A	3E	69
6B	3F	6A
6C	40	6B
6D	41	6C
6E	42	6D
6F	43	6E
70	44	6F
71	45	70
72	46	71
73	47	72
74		



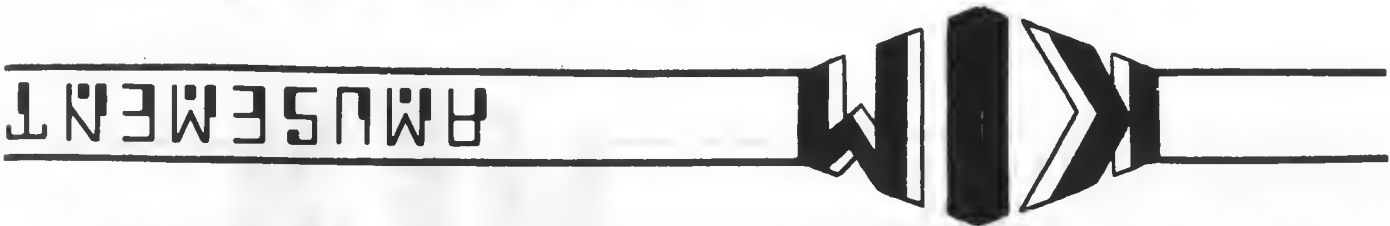
zetten adres TAB1 TAB2 TAB3

c4xd5	A7	5A	23	A8	c1-b2	DA	42	09	DB
c6-d4	A8	6A	1B	A9	e8-08	DB	7C	3E	DC
f3xd4	A9	55	1B	AA	f1-e2	DC	45	0C	DD
c5xd4	AA	62	1B	AB	c8-04	DD	7A	1E	DE
e2-e3	AB	4C	1C	AC	e1-01	DE	44	06	DF
d4-b6	AC	5B	29	AD	c7-c6	DF	72	2A	EO
01-f3	AD	46	15	AE	a2-a4	EO	48	18	E1
08-f6	AE	7E	2D	AF	b8-d7	E1	79	33	00
d2-d4	AF	4B	1B	BO	c1-b2	E2	42	09	E3
c5xd4	BO	62	1B	B1	f7-f6	E3	75	2D	E4
f3xd4	B1	55	1B	B2	e2-e4	E4	4C	1C	E5
e7-e6	B2	74	2C	B3	f8xb4	E5	7D	19	E6
b1-c3	B3	41	12	B4	f1-c4	E6	45	1A	E7
f8-b4	B4	7D	19	B5	08-e7	E7	7E	34	E8
d4-b5	B5	5B	21	B6	f2-f4	E8	4D	1D	E9
e8-08	B6	7C	3E	B7	d7-d5	E9	73	23	EA
c1-f4	B7	42	1D	B8	e4xd5	EA	5C	23	EB
b8-c6	B8	79	2A	B9	b4-d6	EB	59	2B	EC
f4-d6	B9	5D	2B	BA	f4xe5	EC	5D	24	ED
b4xd6	BA	59	2B	BB	f6xe5	ED	6D	24	EE
b5xd6	BB	61	2B	BC	d1-15	EE	43	27	EF
f6-e8	BC	6D	3C	BD	e7-06	EF	74	2E	FO
a2-a3	BD	48	10	BE	01-f3	FO	46	15	F1
b7-b6	BE	71	29	BF	b8-d7	F1	79	33	F2
01-f3	BF	46	15	CG	e1-01	F2	44	06	F3
e5-e4	CG	64	1C	CI	e8-08	F3	7C	3E	00
f3-d4	CI	55	1B	C2	c1-b2	F4	42	09	F5
b8-c6	C2	79	2A	C3	e7-e6	F5	74	2C	F6
e2-e3	C3	4C	14	C4	b4-b5	F6	59	21	F7
c6xd4	C4	6A	1B	C5	a7-a6	F7	70	28	F8
e3xd4	C5	54	1B	C6	a2-a4	F8	48	18	F9
d8-f6	C6	7B	2D	C7	a6xb5	F9	68	21	FA
d4-d5	C7	5B	23	C8	a4xb5	FA	58	21	FB
f8-c5	C8	7D	22	C9	a8xa1	FB	78	00	FC
d1-e2	C9	43	0C	CA	b2xa1	FC	49	00	FD
f6-e7	CA	6D	34	CB	d7-d5	FD	73	23	FE
b1-c3	CB	41	12	CC	01-f3	FE	46	15	FF
08-f6	CC	7E	2D	CD	f8-e7	FF	7D	34	00
d2-d3	CD	4B	13	CE					
e4xc3	CE	5C	13	CF					
e2xe7+	CF	4C	34	DO					
c5xe7	DO	62	34	00					
d7-d5	D1	33	23	D4					
e7-e5	D2	34	24	B2					
08-f6	D3	7E	2D	F4					
01-f3	D4	46	15	D5					
e2-e3	D6	7E	2D	D6					
07-06	D7	76	2E	D8					
c2-c4	D8	1A	1A	DA					
f8-07	D9	36	36	DA					

Zeiten notfallig volgens schakprogramma T.Kortekaas.

xx staan in de aanwijstabel  
zetten staan in "CODE" in TAB1 en TAB2

[illegible]



Programma-deel, schakprogramma TK, op de band zetten om snel een nieuwe partij te kunnen gaan spelen.

INITIALISERING "WIT" KIM "ZWAAR"

PTR	SRØPEN	:7777 00		(v66t "GO")		:7777 01	
		56	4E	4E	4E	4E	4E
INIT4	JMP INI2	0000	4C	26	00		
	LDY X'00'	03	A0	00			
	LDX X'1F'	05	A2	1F			
	STY BORD+16(X)	07	94	26			
	DEX CA	09	CA				
	BPL INI6	0A	10	FB			
	STY ROCO	0C	84	0A			
	STY WRDE	0E	84	0B			
	STX WIS	10	86	90			
	JMP BRK	12	4C	80	17		
		15	FF				
		16	84	83	85	86	82
INIT2	(deel tab.BORD)	1D	80	80	80	80	80
	LDX X'EA'	26	A9	EA			
	STX INI1	28	85	00			
	LDX X'85'	2A	A9	85			
	STA INI1+1	2C	85	01			
	NOP	2E	EA				
	NOP	2F	EA				
	NOP	30	EA				
	LDX X'05'	31	A2	05			
	LDA STOP(X)	33	B5	3E			
	STA VECT(X)	35	9D	FA	17		
	DEX CA	38	CA				
INIT1	BPL INI4	39	10	FB			
	JMP INI5	3B	4C	03	00		
	(deel tab.BORD)	3E	00	00	00	00	00
		46	C1	C1	C1	C1	C1
		4E	C4	C3	C5	C6	C2
		56	C4	C3	C5	C6	C2

Zonodig "Rokade" uitschakelen, adres 000A FF  
KIM-1e zet laten doen, daarna probleem opzetten, zie veld  
adressen, tab.BORD., mogelijk maken zwart de eerste zet te doen,  
zodat KIM kan antwoorden voor Wit.  
Indien nodig "MAX1" en "MAX2", verhogen!  
LDX X'02' 1790 A2 02 (02 wordt 03)  
LDX X'03' 1794 A2 03 (03 " 04)  
KIM rekent daardoor meerdere zetten diep!!!  
Attentie = nog niet volledig uitgetest!!

Met dank aan de Theo Kortekaas, die dit schakprogramma heeft ontwikkeld.  
Leiden, 20 september 1980.  
Jo Wervooren.

BTJ:01 BTJ moet het adres van 01 worden ingevuld.  
BTJ:02 BTJ moet het adres van 02 worden ingevuld.  
Deze routine moet bij voorkeur gebruikt worden  
in samenhang met openingen-routine.

JSR	Q1	0202	20	EA	Q1
NOP		0205		EA	
JMP	Q2	026E	4C	Q2	Q2
NOP		0271		EA	

Voor XX XX moet het adres van TABZWRD worden ingevuld.

Address	Instruction	Comments
0000	LDA X'00'	
0001	STA ZWRD	
0002	LDO NIVO	
0003	LDA X'08'	
0004	STA TABZWRD(X)	
0005	RTS	
0006	CMP X'41'	
0007	BEQ CMPB144	
0008	BIT CZA	
0009	BVS RET	
0010	LDO NIVO	
0011	STA TABZWRD(X)	
0012	CMP X'8F'	
0013	BEQ RET	
0014	CA F0	
0015	CA F0	
0016	CA F0	
0017	CA F0	
0018	CA F0	
0019	CA F0	
0020	CA F0	
0021	CA F0	
0022	CA F0	
0023	CA F0	
0024	CA F0	
0025	CA F0	
0026	CA F0	
0027	CA F0	
0028	CA F0	
0029	CA F0	
0030	CA F0	
0031	CA F0	
0032	CA F0	
0033	CA F0	
0034	CA F0	
0035	CA F0	
0036	CA F0	
0037	CA F0	
0038	CA F0	
0039	CA F0	
0040	CA F0	
0041	CA F0	
0042	CA F0	
0043	CA F0	
0044	CA F0	
0045	CA F0	
0046	CA F0	
0047	CA F0	
0048	CA F0	
0049	CA F0	
0050	CA F0	
0051	CA F0	
0052	CA F0	
0053	CA F0	
0054	CA F0	
0055	CA F0	
0056	CA F0	
0057	CA F0	
0058	CA F0	
0059	CA F0	
0060	CA F0	
0061	CA F0	
0062	CA F0	
0063	CA F0	
0064	CA F0	
0065	CA F0	
0066	CA F0	
0067	CA F0	
0068	CA F0	
0069	CA F0	
0070	CA F0	
0071	CA F0	
0072	CA F0	
0073	CA F0	
0074	CA F0	
0075	CA F0	
0076	CA F0	
0077	CA F0	
0078	CA F0	
0079	CA F0	
0080	CA F0	
0081	CA F0	
0082	CA F0	
0083	CA F0	
0084	CA F0	
0085	CA F0	
0086	CA F0	
0087	CA F0	
0088	CA F0	
0089	CA F0	
0090	CA F0	
0091	CA F0	
0092	CA F0	
0093	CA F0	
0094	CA F0	
0095	CA F0	
0096	CA F0	
0097	CA F0	
0098	CA F0	
0099	CA F0	
0100	CA F0	

Aanpassingen KIM-schaakprogramma T.Kortekaas.  
 Routine "Quick", kan overal in het geheugen  
 worden geplaatst.





0130:	1000	05	F1	NXT00	LDA	AOL
0140:	1002	05	F3		CMP	ATL
0150:	1004	05	F2		LDA	AOL
0160:	1006	05	F4		SBC	ATL
0170:	1008	05	F1		INC	AOL
0180:	1009	00	02		BNE	RTSFB
0190:	100C	06	F2		INC	AOL
0200:	100E	60			RTSFB	RTS
0210:	100F	00	4B	HEADR	LDYIM	\$4B
0220:	1011	20	21	10	JSR	ZERDLY
0230:	1014	00	F9		BNE	HEADR
0240:	1016	05	F6		ADCIM	\$FE
0250:	1018	00	F3		BOS	HEADR
0260:	1019	00	21		LDYIM	\$21
0270:	101C	00	01	10	WRBIT	JSR
0280:	101E	00			INX	
0290:	1020	00			INX	
0300:	1021	00			ZERDLY	DEY
0310:	1022	00	FD		BNE	ZERDLY
0320:	1024	00	25		BOS	WRTPPE
0330:	1025	00	00		LDYIM	\$32
0340:	1028	00			ONEDLY	DEY
0350:	1029	00	FD		BNE	ONEDLY
0360:	102B	40			WRTPPE	PHQ
0370:	102C	00	42	17	LDR	0BD
0380:	102F	40	00		EDPIM	\$80
0390:	1031	00	42	17	STB	0BD
0400:	1034	00			PLB	
0410:	1035	00	20		LDYIM	\$20
0420:	1037	00			DEX	
0430:	1038	00			RTS	
0440:	1039	02	06		RDBYTE	LDXIM
0450:	103B	40			ROBYTT	PHQ
0460:	103C	20	47	10	JSR	RDTHIT
0470:	103F	00			PLB	
0480:	1040	20			ROLB	
0490:	1041	00	30		LDYIM	\$30
0500:	1043	00			DEX	
0510:	1044	00	F5		BNE	ROBYTT
0520:	1045	00			RTS	
0530:	1047	20	40	10	RDTHIT	JSR
0540:	1049	00			DEY	RDTHIT
0550:	104B	00	42	17	LDB	TAPBIN
0560:	104E	40	F5		EOR	LDSTIN
0570:	1050	10	F8		BPL	RDBIT
0580:	1052	40	F5		EOR	LDSTIN
0590:	1054	05	F5		STB	LDSTIN
0600:	1056	00	80		COPYIM	\$80
0610:	1058	60			RTS	



APPLE MICRO-WARE assembler SSX-1.0 PAGE 23

0520: 1055 D8	WRITE	CLD			
0530: 1059 A9 BF		LDAIM #BF			
0540: 105C 8D 43 17		STB PBD			
0550: 105F A9 27		LDAIM #27			
0560: 1061 8D 42 17		STB PBD			
0570: 1064 A9 40		LDAIM #40			
0580: 1066 23 2F 12		JSR HDRR			
0590: 1069 A9 27		LDAIM #27			
0600: 106B 92 00		LDAIM #00			
0610: 106D 41 F1		BCR ROL			
0620: 106F 48		PHB			
0630: 1070 81 F1		LDAIX ROL			
0640: 1072 20 84 10		JSR WRBYTE			
0650: 1075 20 02 10		JSR NXT90			
0660: 1078 60 1D		LDAIM #1D			
0670: 1079 60 33		PLB			
0680: 107B 60 3E		BCC WRD			
0690: 107D A9 22		LDAIM #22			
0700: 107F 20 84 10		JSR WRBYTE			
0710: 1082 50 47		BCB BEL			
0720: 1084 02 10		LDAIM #10			
0730: 1086 00 50		WBYT			
0740: 1087 10 12 10		JSR WBYT			
0750: 1089 00 50		BNE WBYT			
0760: 108C 50		RTS			
0770: 108D 08		CLD			
0780: 108F 20 47 10		JSR RDTBIT			
0790: 1091 95 1E		LDAIM #1E			
0800: 1093 20 05 10		JSR HQRDR			
0810: 1095 85 F0		STB CHKSUM			
0820: 1098 20 67 10		JSR RDTBIT			
0830: 109B A0 24		LDAIM #24			
0840: 109D 20 48 10		JSR RDBIT			
0850: 109F 80 F9		BCB RDTWC			
0860: 10A2 20 40 10		JSR RDBIT			
0870: 10A5 80 2B		LDAIM #2B			
0880: 10A7 20 39 10		JSR RDTM4			
0890: 10A9 81 F1		RTPIX ROL			
0900: 10AB 45 F0		BCB CHKSUM			
0910: 10AD 85 F0		STB CHKSUM			
0920: 10B0 20 00 10		JSR NXT90			
0930: 10B3 00 25		LDAIM #25			
0940: 10B5 90 F2		BCC RDTM4			
0950: 10B7 20 39 10		JSR RDBYTE			
0960: 10BA C5 F0		BCB CHKSUM			
0970: 10BD F0 0D		BCB BEL			
0980: 10BE 89 45		LDAIM #45			
0990: 10C0 20 30 02		JSR OUTPUT			
1000: 10C3 A9 52		LDAIM #52			
1010: 10C5 20 30 03		JSR OUTPUT			
1020: 10C8 20 30 03		JSR OUTPUT			
1030: 10CB 60		BEL			
1040: 10CE 60		RTS			



Patches op MICROSOFT BASIC

H.J.C. Otten

Een gemis van Microsoft BASIC ten opzichte van Tiny BASIC (KIM versie) is het ontbreken van berekende regelnummers. Een berekende sprong is vaak erg handig en efficiënt, bovendien kan een subroutine met een naam worden aangeroepen wat de duidelijkheid bevordert. Die naam is dan een (integer) variabele die we van te voren een waarde geven. Een voorbeeld :

10 TIOF = 1000

20 A=1 : B=2

30 GOSUB TIOF

40 PRINT C

:

1000 REM SUBROUTINE TIOF , C = A+B

1010 C = A + B

1020 RETURN

Het gebruiken van getallen en niet van expressies heeft het voordeel dat het snel is. De patch moet de snelheid van een GOTO etc niet aantasten liefst. De volgende patch levert het gewenste resultaat :

patch 27F2 in 4C 00 04 ( of een andere lokatie ; )

zet op 0400 : B0 06 BCS REKEN EERSTE KARAKTER EEN GETAL?

20 B828 JSR VERZAMEL REGELNUMMER IN (11,12)

4C F527 JMP verder met routine (normaal)

REKEN 20 342C JSR GET ADDRESS in FPaccu

20 EF35 JSR CONVERT ADDRESS TO INTEGER IN (11,12)


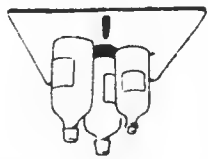




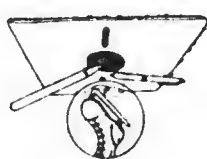



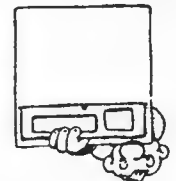
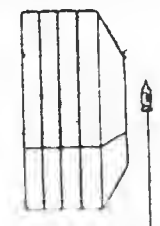
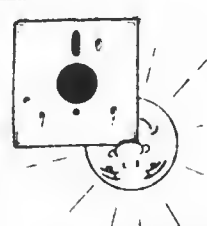
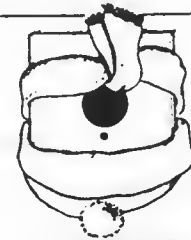
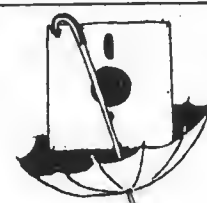
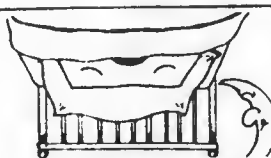
4C F527 JMP verder met routine

Met deze patch zal BASIC naar een regelnummer achter GOTO etc

springen op de oude manier, is het eerste karakter echter geen getal ( na GOTO etc ) dan wordt de expressie berekend en naar

het berekende regelnummer gesprongen.

# AANWIJZINGEN VOOR GEBRUIK EN OPSLAG VAN FLOPPY DISK

<p>1. Raak het oppervlak van de floppy niet met vingers aan; Gevaar op beschadiging, waardoor fouten ontstaan.</p> 	<p>2. Gebruik geen oplosmiddelen zoals alcohol of verdunners om de floppy te reinigen.</p> 	<p>3. Gebruik geen magnetische of statisch geladen voorwerpen vlakbij de floppy. Gegevens op de floppy kunnen verloren gaan, als de floppy aan een magnetisch veld wordt blootgesteld.</p> 	<p>4. De floppy niet buigen of vouwen.</p> 	<p>5. Plaats geen zware voorwerpen boven op de floppy.</p> 	<p>6. Schuif geen klemmen of paper-clips aan de floppy.</p> 	<p>7. Beschrijf het etiket op de floppy nooit met een potlood of balpen. Gebruik een viltstift.</p> 	<p>8. Gebruik geen gum.</p> 	<p>9. Plaak het identificatie-etiket op de juiste plaats. Plaak nooit etiketten in lagen over elkaar heen.</p> 	<p>10. Floppy aan bovenrand vastpakken en voorzichtig in het aandringmechanisme stoppen.</p> 	<p>11. Stop de floppy na gebruik altijd onmiddellijk terug in de bijbehorende enveloppe.</p> 	<p>12. Bewaar de floppy's rechtop in de bijbehorende doos.</p> 	<p>13. Stel de floppy nooit bloot aan overmatige warmte, verhitting of zonlicht.</p> 	<p>14. Omgevingstemperaturen voor gebruik: 10° C tot 50° C en vochtigheidsgraad: 20% RH tot 80% RH.</p> 	<p>15. Omgevingstemperaturen voor opslag: 4° C tot 53° C en vochtigheidsgraad: 8% RH tot 80% RH.</p> 	<p>16. Tijdens transport dient de floppy in de bijbehorende enveloppe en een beschermende doos te zijn opgeborgen. Omgevingsstemperatuur: -40° C tot +53° C en vochtigheidsgraad: 8% RH tot 90% RH.</p> 
--	---	--	--	---	---	--	--	--	---	--	--	--	---	--	---